



HELMUT SCHMIDT  
UNIVERSITÄT

Universität der Bundeswehr Hamburg

# **Automatisierte Bestimmung von Eigenschaften industrieller Produktionssysteme unter Einfluss evolutionärer Änderungen**

Von der Fakultät für Maschinenbau

der Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg

zur Erlangung des akademischen Grades eines Doktor-Ingenieurs

genehmigte

DISSERTATION

Vorgelegt von

Jan Ladiges, M.Sc.

aus Hamburg

Hamburg 2018

Gutachter: Prof. Dr.-Ing. Alexander Fay  
Prof. Dr. Winfried Lamersdorf  
Vorsitzender: Prof. Dr.-Ing. habil. Thomas Klassen  
Tag der mündlichen Prüfung: 31.08.2018

## Vorwort des Autors

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, der ich zwischen Oktober 2012 und Mai 2017 nachging. Entstanden ist die Arbeit aus meinen Ergebnissen des Forschungsprojektes „Forever Young Production Automatio with Active Components“ als Teil des DFG-Schwerpunktprogrammes „SPP1593 – Managed Software Evolution“.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing Alexander Fay. Neben der Gelegenheit mir an seiner Professur als wissenschaftlicher Mitarbeiter den Wunsch zu erfüllen, forschend im Bereich der Automatisierungstechnik tätig zu sein, stand er mir auch stets mit seiner Fachkenntnis und methodischen Herangehensweise zur Seite. Die Gelegenheiten an verschiedenen aktuellen Forschungsthemen mit ständigem Bezug zur industriellen Praxis zu arbeiten habe ich mit Freude und Einsatz angenommen. Die fachlichen Diskussionen, der Rat, die nötige Fokussierung sowie die Förderung und Forderung durch ihn gehören zu den wesentlichen Faktoren des Erfolges dieser Arbeit.

Herrn Prof. Dr. Winfried Lamersdorf danke ich nicht nur für die Begutachtung meiner Arbeit, sondern auch für die mit Herrn Prof. Dr.-Ing Alexander Fay gemeinsam durchgeführte Betreuung des o.g. Forschungsprojektes. Die freundliche und ruhige Art von Herrn Prof. Dr. Lamersdorf sowie seine Erfahrung und methodischen Kenntnisse machten ihn stets zu einem angenehmen Diskussionspartner, von dem ich viel lernen konnte.

Ich danke Herrn Prof. Dr.-Ing. habil. Thomas Klassen für die Übernahme des Prüfungsvorsitzes.

Ebenfalls großen Dank möchte ich an Herrn Christopher Haubeck aussprechen, mit dem ich gemeinsam das Forschungsprojekt durchgeführt habe. Viele der in dieser Arbeit niedergeschriebenen Ansätze fanden ihren Ursprung in einer der vielen interessanten und zielführenden Diskussionen mit ihm. Ebenfalls bedanken möchte ich mich bei den Herren Ireneus Wior, Prof. Dr. Lars Braubach und Dr. Alexander Pokahr, die mit Ihrer Arbeit am Forschungsantrag den Grundstein für diese Arbeit legten.

Besonderer Dank gilt allen Kollegen des Instituts für Automatisierungstechnik, die mir im Laufe der Zeit mit ihrer Expertise und fachlichem Rat zur Seite standen. Die Zusammenarbeit war stets freundschaftlich geprägt und fand in einer für mich sehr angenehmen Atmosphäre statt. Der Teamzusammenhalt im Institut war stets ein Erfolgsgarant für die Arbeiten, die wir durchgeführt haben. Besonders hervorheben möchte ich diejenigen Kollegen, die sich bereit erklärten mich durch Korrekturlesen und in der Vorbereitung der Verteidigung unterstützten: Herr Henry Bloch, Herr Luu Hoang, Herr Dr.-Ing. Maik Riedel und Herr Dr.-Ing. Andre Scholz. Des Weiteren möchte ich mich bei Herrn Dr.-Ing. Thomas Holm und Herrn Aljosha Köcher für die enge Zusammenarbeit und bei Herrn Dr.-Ing. Esteban Arroyo für drei intensive Jahre im gemeinsamen Büro mit vielen fruchttragenden Diskussionen bedanken.

Zudem möchte ich Frau Sarah Dinse-Pötsch für das Korrekturlesen der Arbeit danken.

Insbesondere bedanke ich mich bei meiner Frau Yvonne. Sie stand mir in der gesamten Zeit stets zur Seite und stärkte mir in den Jahren immer den Rücken. Gerade in den stressigen Zeiten, die nicht immer leicht waren, war sie mein Ruhepol und mein mentales Rückgrat, dies möchte ich besonders würdigen.

Ich widme diese Arbeit meinem Sohn Phil, meiner Großmutter Wilma und meinem Onkel Jürgen.

## Inhaltsverzeichnis

Vorwort des Autors .....	I
Inhaltsverzeichnis.....	II
Abkürzungsverzeichnis.....	V
1 Einleitung.....	1
1.1 Motivation und Hintergrund.....	1
1.2 Zielsetzung und Aufbau der Arbeit .....	2
2 Eingrenzung der betrachteten Systeme und vorausgehende Untersuchungen .....	5
2.1 Eingrenzung und Beschreibung der betrachteten Systeme .....	5
2.1.1 Zum Begriff der Produktion.....	5
2.1.2 Die Produktionsanlage als mechatronisches System.....	6
2.1.3 Unterscheidung in diskrete Fertigung und verfahrenstechnische Produktion .....	7
2.1.4 Weitere Eingrenzung und Annahmen.....	8
2.1.5 Formale Beschreibung.....	9
2.2 Vorgehen im Engineering von Produktionssystemen und ihrer Evolution.....	10
2.3 Evolutionäre Änderungen von Produktionssystemen .....	11
2.3.1 Begriffsdefinitionen.....	11
2.3.2 Evolutionstreiber eines Produktionssystems.....	12
2.3.3 Unterscheidung von Evolutionsszenarien.....	14
3 Nicht-funktionale Eigenschaften zur Bewertung evolvierender Fertigungsanlagen.....	20
3.1 Begriffsklärung zu nicht-funktionalen Anforderungen an Fertigungsanlagen .....	20
3.2 Bewertung von Produktionssystemen mittels Messung nicht-funktionaler Eigenschaften .....	22
3.2.1 Beschreibung der Bewertung anhand von Merkmalen .....	22
3.2.2 Ermittlung der Ausprägung von Qualitätsmerkmalen .....	24
3.3 Kennzahlen und Metriken nicht-funktionaler Eigenschaften .....	26
3.3.1 Performance im Sinne von Zeit- und Mengen-bezogenen Leistungskennzahlen.....	27
3.3.2 Verfügbarkeit und Zuverlässigkeit .....	33
3.3.3 Flexibilität.....	35
4 Beschreibungsmittel zur Verhaltensmodellierung von Fertigungsanlagen .....	40
4.1.1 Zustandsautomaten .....	40
4.1.2 Petri-Netze .....	41
4.1.3 UML und SysML.....	42
4.1.4 Algebraische Modelle.....	43
4.1.5 Process Specification Language .....	43
5 Bewertung evolvierender Fertigungssysteme – Stand der Forschung und industrielle Praxis.....	44

---

5.1	Modellbasierte Ansätze .....	44
5.1.1	Modellbasierte Analyse und Bewertung von Fertigungssystemen.....	45
5.1.2	Kritische Betrachtung.....	48
5.2	Modellernende Ansätze.....	49
5.2.1	Modellgenerierung im Rahmen der Fehlerdiagnose .....	50
5.2.2	Process Mining .....	53
5.2.3	Kritische Betrachtung.....	54
5.3	Industrielle Praxis.....	55
6	Forschungsbedarf, Anforderungen und These.....	57
7	Allgemeiner Ansatz zur Wissensextraktion aus Ereignissequenzen.....	61
7.1	Wissensarchitektur .....	62
7.2	Datenakquirierung .....	63
7.3	Informationsmodellierung.....	64
7.4	Auswahl des Beschreibungsmittels.....	67
8	Maschinenzustandsmodelle: Definition, Lernalgorithmus und Anomalieerkennung.....	69
8.1	Definition .....	69
8.2	Preprocessing der Ereignissequenz .....	72
8.3	Generieren des Petri-Netzes.....	73
8.3.1	Generieren des auf Maschinenverhalten bezogenen Teils.....	73
8.3.2	Generieren des auf Signalverhalten bezogenen Teils.....	77
8.4	Beispiel.....	78
8.5	Anomalieerkennung.....	82
9	Materialflussmodelle: Definition, Lernalgorithmus und Anomalieerkennung .....	85
9.1	Definition .....	85
9.2	Preprocessing der Ereignissequenz .....	89
9.2.1	Aufteilung und Filtrierung der Ereignisse.....	90
9.2.2	Kombinieren von einzelnen Ereignissen zu Materialfluss-relevanten Ereignissen.....	90
9.2.3	Lokale Zuweisung von Ereignissen zu Werkstückinstanzen.....	92
9.2.4	Verknüpfung lokaler Zuweisungen zu anlagenweiten globalen IDs .....	100
9.3	Generieren des Petri-Netzes.....	104
9.4	Anomalieerkennung.....	109
10	Eigenschaftsbestimmung.....	112
10.1	Ermittlung der Positionen zur Werkstückbearbeitung .....	112
10.2	Zeiten und zeitbezogene Größen.....	113
10.3	Produzierte Menge .....	117

---

10.4	Topologie- und Strukturgrößen .....	117
10.5	Planungs- und Fähigkeitsgrößen.....	118
10.6	Zusammenfassung zur Eigenschaftsbestimmung.....	118
11	Fallstudienbasierte Evaluierung des Ansatzes.....	121
11.1	Anwendung auf eine Pick and Place Unit .....	121
11.1.1	Aufbau der Anlage und betrachtete Evolutionsszenarien .....	122
11.1.2	Durchführung der Fallstudie .....	124
11.1.3	Diskussion der Ergebnisse .....	126
11.2	Anwendung auf eine Job-Shop Fertigung.....	128
11.2.1	Aufbau der Anlage und betrachtete Evolutionsszenarien .....	129
11.2.2	Durchführung der Fallstudie .....	132
11.2.3	Diskussion der Ergebnisse .....	133
12	Bewertung, Zusammenfassung und Ausblick.....	135
12.1	Bewertung und Grenzen des Ansatzes .....	135
12.2	Zusammenfassung .....	137
12.3	Weiterer Handlungsbedarf .....	139
	Anhang A: Verwendete Formelzeichen und Notation .....	142
	Anhang B: Notation der Unified Modeling Language .....	147
	Anhang C: Zusammenfassende Darstellung der Messelemente für Qualitätsmerkmale.....	149
	Anhang D: Ereignissequenzen der Beispiele .....	151
	Anhang E: Daten und Ergebnisse der Anwendung auf der „Pick and Place Unit“ .....	162
	Anhang F: Daten und Ergebnisse zur Anwendung auf der Job-Shop Fertigung.....	170
	Literaturverzeichnis.....	188
	Referenzierte Normen, Richtlinien und Empfehlungen.....	207
	Referenzierte Internetquellen.....	209
	Veröffentlichungen des Verfassers .....	210
	Studentische Arbeiten.....	212

## Abkürzungsverzeichnis

B/E-Netz	Bedingungs-Ereignis-Netz
DEA	Efficiency Data Envelopment Analysis
DES	Discrete Event System
ERP	Enterprise Ressource Planning
FPY	First Pass Yield
HiL	Hardware-in-the-Loop
HSU	Helmut-Schmidt-Universität
IEC	International Electrotechnical Commission
I/O	Input/Output (Ein- und Ausgang einer SPS)
ISO	International Organization for Standardization
KPI	Key Performance Indicator
MBE	Model Based Engineering
MBT	Modell-basiertes Testen
MDE	Model Driven Engineering
MES	Manufacturing Execution System
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NFA	Nicht-funktionale Anforderung
NFE	Nicht-funktionale Eigenschaft
OEE	Gesamtanlageneffektivität (Overall Equipment Effectiveness)
PLS	Prozessleitsystem
PLT	Prozessleittechnik

---

PPU	Pick and Place Unit
PSL	Process Specification Language
RFID	Radio-frequency identification
SCADA	Supervisory Control and Data Acquisition
SIPN	Signal interpretierbare Petri-Netze
SPS	Speicherprogrammierbare Steuerung
SQuaRE	Systems and Software Quality Requirements and Evaluation
SWOT	Strengths, Weaknesses, Opportunities, Threats
SysML	System Modeling Language
UML	Unified Modeling Language
VDI	Verein Deutscher Ingenieure
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
WZM	Werkzeugmaschine

# 1 Einleitung

## 1.1 Motivation und Hintergrund

Aufgrund der hohen Investitionskosten für Produktionssysteme werden diese, je nach Branche, üblicherweise viele Jahre bis hin zu mehreren Dekaden betrieben [ZVE10]. In dieser langen operativen Phase wird eine kontinuierliche Optimierung des Produkt- und Portfoliowertes als ein Handlungsansatz zur Erhaltung der Absatzsteigerung trotz der Erwartung eines steigenden Wettbewerbs gesehen [VDMc14]. Zur Optimierung gehören nicht nur stetig angepasste Produktionsprozesse, sondern auch die Individualisierung der Produkte. Allgemein zeichnet sich ein Trend Richtung „mass customization“, kleinen Losgrößen, hoher Variabilität von Produkttypen sowie sich ändernder Portfolios während des Lebenszyklus der immer komplexeren Produktionssysteme ab [VDMc14; VFS+15]. Daher gehört schon heute die Einführung neuer Produkte oder Produktfamilien zu den häufigsten Gründen von Änderungen an Produktionssystemen [BESÄ10]. Aber auch neue Umweltauflagen oder Technologieentwicklungen führen dazu, dass Firmen der verarbeitenden Industrie sich langfristig nur behaupten, wenn sie sich konstant an ändernde Bedingungen anpassen [Wes03]. Folglich unterliegen die Produktionssysteme in ihrer langen operativen Phase häufig Änderungen in den an sie gestellten Anforderungen [VFS+15]. Entsprechend steigt der Bedarf an Wandlungsfähigkeit produzierender Systeme kontinuierlich [ScWi04].

Folgerichtig hat eine Änderungsmanagement-Studie von 2012 [LWM+12] ergeben, dass technische Änderungen ein Viertel der gesamten Forschungs- und Entwicklungskapazität deutscher Unternehmen in Anspruch nehmen<sup>1</sup>. Produktionssysteme sind somit nicht als starre, sondern eher als sich stetig ändernde Systeme zu betrachten. Allgemein kann davon ausgegangen werden, dass Produktionssysteme einer kontinuierlichen Evolution während ihrer Betriebsphase unterliegen. Entsprechend sind Änderungen an solchen Systemen als Regelfall und nicht als Ausnahmeerscheinung zu betrachten [TOM10; VFS+15].

Die üblicherweise von der Literatur vorgeschlagene Vorgehensweise zur Durchführung von Änderungen ist das Durchlaufen eines systematischen Re-Engineering Prozesses. Dieser besteht aus (1) einer Formalisierung neuer oder geänderter Anforderungen, (2) dem Design einer Systemlösung (durch Änderung des aktuell vorhandenen Designs), idealerweise mithilfe eines modellbasierten Ansatzes, (3) der Validierung der Lösung gegen die Anforderungen und (4) schlussendlich der Umsetzung der geplanten Änderung auf dem realen System [BESÄ10]. In der industriellen Praxis lässt sich jedoch häufig eine andere Vorgehensweise beobachten. Nicht selten werden die Formalisierung von Anforderungen und die Vorab-Bewertung möglicher Systemlösungen ausgelassen, und stattdessen werden informelle Anforderungen direkt im System umgesetzt [BESÄ10; FRLi00; HuMA99; VFS+15]. Als Hauptgrund hierfür kann der hohe Zeit- und Kostendruck insbesondere bei der Durchführung kleinerer Änderungen genannt werden [BESÄ10]. Aber auch fehlendes methodisches Wissen des Anlagenpersonals spielt hier eine Rolle, vgl. [ScFA13]. Diese Umstände führen dazu, dass insbesondere kleinere unerwartete Änderungen, die kurzfristig umgesetzt werden, unzulänglich dokumentiert und bewertet werden [VFS+15]. Im Falle der Bewertung trifft dies besonders auf die Betrachtung nicht-funktionaler Eigenschaften zu.

Offensichtlich beinhaltet diese Vorgehensweise Risiken und Nachteile. Das geänderte Verhalten der Systeme ist ungenügend dokumentiert. Folglich ist explizites Prozesswissen nicht verfügbar. Somit ist das dokumentierte Anlagenverhalten unvollständig oder widerspricht sogar dem aktuellen Verhalten

---

<sup>1</sup> Hier sind allerdings auch Änderungen in laufenden Entwicklungsprojekten, also vor der operativen Phase der Anlagen, inbegriffen.

des realen Systems. Diese Diskrepanz zwischen explizitem Anlagenwissen und der aktuellen Realisierung des Systems kann als *Wissenslücke* bezeichnet werden [HWB+13<sup>#</sup>]. Darüber hinaus weisen Produktionssysteme eine sehr hohe Komplexität aufgrund der Vielzahl heterogener interagierender Komponenten auf. Änderungen betreffen häufig mehrere der in Produktionssystemen vorkommenden Disziplinen Maschinenbau, Elektrotechnik und Software [VFS+15]. Die Wissenslücke zu schließen ist daher eine aufwändige und fehleranfällige Aufgabe. Die Bewertung geänderten Verhaltens beinhaltet auch die Bewertung der komplexen Interrelationen zwischen den Disziplinen [BBO+12]. Die hohe Komplexität führt dazu, dass die Zusammenhänge zwischen den getätigten Änderungen und Performance-Größen sowie anderen Merkmalsausprägungen der Systeme nicht bekannt sind. Häufig ist der Zusammenhang zwischen den Disziplinen nicht einmal analytisch beschreibbar [CHR06]. Es muss zudem davon ausgegangen werden, dass Eigenschaften der Systeme über die Disziplinen hinweg durch umgesetzte Änderungen beeinflusst werden [WIT12]. Die Erkennung solcher Seiteneffekte ist ebenfalls kompliziert und zeitaufwändig [VFS+15], insbesondere unter Berücksichtigung fehlender Dokumentation vorangegangener Änderungen. Daraus resultiert, dass Eigenschaften der evolvierenden Systeme u.U. ungenügend ausgeprägt sind und solche unnötigen Suboptimalitäten zudem vom Anlagenpersonal nicht erkannt werden. Daher gehört die Sicherstellung einer Konsistenz zwischen Dokumentation und Verhalten zur Erhaltung der Qualität einer Anlage zu den wichtigsten Herausforderungen innerhalb der Forschung zu evolvierenden Produktionssystemen [VFS+15].

## 1.2 Zielsetzung und Aufbau der Arbeit

Betrachtungsgegenstand der vorliegenden Arbeit ist die diskrete Fertigung. Das Ziel der Arbeit liegt in der Erarbeitung einer Methode, die o.g. Defiziten bzgl. fehlender aktueller Dokumentation und Bewertung evolvierender Fertigungssysteme entgegenwirkt. Dabei werden die in der industriellen Praxis vorherrschenden Rahmenbedingungen berücksichtigt. Dies umfasst in erster Linie die Reduktion des manuellen Aufwandes zur Dokumentation und Bewertung sich ändernden Verhaltens sowie ein möglichst geringes Maß an notwendigen methodischen Kenntnissen zur Anwendung der Methode. Als weitere wichtige Nebenbedingung des zu erarbeitenden Ansatzes sei hier genannt, dass idealerweise der Produktionsbetrieb durch die Anwendung des Ansatzes nicht beeinflusst wird. Dies schließt u.a. auch den Zugriff auf Steuerungscode oder interne Variablen der Systeme aus, um eine Einwirkung auf die Ausführungsgeschwindigkeit im laufenden Betrieb zu vermeiden. Somit müssen die für den Ansatz zu verwendenden Daten über externe Schnittstellen verfügbar sein.

Zur Bewertung der Evolution werden die Möglichkeiten einer weitestgehend automatischen Ermittlung der Ausprägungen von Prozess- und Anlageneigenschaften sowie einer Erkennung von Änderungen dieser untersucht. Insbesondere wird die automatisierte Modellgenerierung auf Basis von im Betrieb der Systeme generierten dynamischen Daten betrachtet. Die resultierenden Modelle dienen als Dokumentationsartefakte des evolvierenden Verhaltens. Zudem müssen sie herangezogen werden können, um, idealerweise ebenfalls automatisiert, Analysen hinsichtlich verschiedener Eigenschaften des Systems zur Bewertung vorgenommener Änderungen vornehmen zu können. Dabei stehen insbesondere solche Eigenschaften im Fokus, die möglichst generisch für die Bewertung einer Vielzahl an Fertigungssystemen herangezogen werden können und die üblicherweise auch für Anlagenbetreiber von Interesse sind. Das heißt, die Eigenschaften müssen abstrakt und aus einschlägiger Literatur sowie Normen, Richtlinien etc. bekannt sein.

Von wissenschaftlicher Relevanz ist dabei die Untersuchung des hier skizzierten Vorhabens unter den gegebenen Nebenbedingungen. Hierzu gehört die Klärung, welche Eigenschaften unter den vorherr-

schenden Umständen ermittelbar sind und welche a priori Informationen notwendig sind, um geeignete Analyseergebnisse zu erhalten. Zudem müssen Annahmen, Realisierungsmöglichkeiten und Grenzen eines solchen Ansatzes geklärt werden. Konkret bedarf es hierzu entsprechender Modellierungen und geeigneter Algorithmen. Folgende Forschungsfragen werden daher adressiert:

- a) Unter welchen Bedingungen können über Beobachtungen extern verfügbarer Daten Verhaltensmodelle erlernt werden?
- b) Inwieweit ist es möglich, einen Zusammenhang zwischen aus Daten generierten Verhaltensmodellen und aus der Literatur bekannten Eigenschaften fertigungstechnischer Systeme herzustellen?
- c) Ist es somit möglich, Evolution zu erkennen und zu bewerten?
- d) Welche zusätzlichen Informationen sind hierfür notwendig?
- e) Wie können diese einfach und schnell in einen entsprechenden Ansatz integriert werden?
- f) Welche Methoden und Algorithmen werden benötigt und wie sind diese umzusetzen?

Der Aufbau der vorliegenden Arbeit ist in Abbildung 1-1 skizziert. Im Anschluss an diese Einleitung werden zunächst in den Kapiteln 2 bis 4 die notwendigen Grundlagen aufgezeigt. Die Kapitel 2 und 3 zeigen dabei die Rahmenbedingungen bzgl. der betrachteten Systeme und ihrer Eigenschaften auf. Hierzu klärt Kapitel 2 zunächst grundlegende Begriffe, schränkt die betrachteten Systeme ein und trifft notwendige Annahmen. Zudem wird kurz aufgezeigt, welches Vorgehen bei der Durchführung von Änderungen an Produktionssystemen von der Literatur vorgeschlagen wird. Des Weiteren wird die Evolution von Produktionssystemen näher betrachtet. Hierzu gehören insbesondere die Gründe und Treiber für Anlagenevolution, welche Arten von Evolution existieren sowie die Unterscheidung tangierter Disziplinen durch die Änderungen und einer daraus resultierenden Kategorisierung von Evolutionsszenarien. Kapitel 3 beschäftigt sich mit den Eigenschaften, die sich zur Bewertung der Evolution fertigungstechnischer Anlagen eignen. Im Anschluss an allgemeine Begriffsklärungen und Diskussionen zur Bewertung von Produktionssystemen wird insbesondere eine Menge an Eigenschaften aufgezeigt, die zur Bewertung herangezogen werden können. Diese Eigenschaften entstammen einer intensiven Literaturrecherche und sind in Form von Kennzahlen und Metriken dargelegt. Somit stellen sie bereits ein erstes Ergebnis der vorliegenden Arbeit dar. Kapitel 4 beschreibt die Grundlagen der Verhaltensmodellierung von Fertigungsanlagen und diskutiert Beschreibungsmittel, die sich hierfür eignen. Kapitel 5 dient der Darstellung des relevanten Standes der Forschung. Insbesondere werden hier zwei Arten von Ansätzen diskutiert, die sich zur Evolutionsunterstützung unter o.g. Nebenbedingungen eignen. Diese Ansätze lassen sich unterteilen in modellbasierte Verfahren und modelllernende Verfahren. Theoretische Überlegungen sowie eine Betrachtung der aus der Literatur zu entnehmenden industriellen Praxis sollen dabei aufzeigen, inwieweit sich die diskutierten Ansätze zur Erfüllung obiger Zielsetzung eignen. Diese Überlegungen münden in den in Kapitel 6 aufgezeigten Forschungsbedarf.

<b>Einleitung (Kapitel 1)</b>	
<b>Eingrenzung und vorausgehende Untersuchungen (Kapitel 2)</b>	<b>Nicht-funktionale Eigenschaften zur Bewertung evolvierender Fertigungsanlagen (Kapitel 3)</b>
<b>Beschreibungsmittel zur Verhaltensmodellierung von Fertigungsanlagen (Kapitel 4)</b>	
<b>Stand der Forschung und industrieller Praxis zur Evaluierung evolvierender Fertigungssysteme (Kapitel 5)</b>	
<b>Forschungsbedarf, Anforderungen und These (Kapitel 6)</b>	
<b>Allgemeiner Ansatz (Kapitel 7)</b>	
<b>Maschinenzustandsmodelle (Kapitel 8)</b>	<b>Materialflussmodelle (Kapitel 9)</b>
<b>Eigenschaftsbestimmung (Kapitel 10)</b>	
<b>Fallstudien (Kapitel 11)</b>	
<b>Bewertung, Zusammenfassung und Ausblick (Kapitel 12)</b>	

**Abbildung 1-1: Aufbau der vorliegenden Arbeit**

Auf Basis des dargelegten Forschungsbedarfs werden ebenfalls in Kapitel 6 die Anforderungen an den Ansatz sowie eine These, die es zu bestätigen oder zu widerlegen gilt, formuliert. Die Kapitel 7-10 beschreiben den Ansatz zur evolutionsunterstützten Methode. Der allgemeine Ansatz bzw. das Framework des Ansatzes ist dabei in Kapitel 7 beschrieben. Es beinhaltet ebenfalls die Auswahl des in den darauf folgenden Kapiteln verwendeten Beschreibungsmittels. Zur Umsetzung des Frameworks bedarf es verschiedener Algorithmen. Daher werden in den Kapiteln 8 und 9 zwei verschiedene Arten von Modellen beschrieben, die im Rahmen des Ansatzes aus verfügbaren Daten generiert werden. Speziell sind das Maschinenzustandsmodelle sowie Materialflussmodelle. Die Kapitel enthalten die Definition der Modelle, Lernalgorithmen zur Generierung dieser sowie die Definition von Anomalien, die zur Erkennung geänderten Verhaltens herangezogen werden können. Kapitel 10 beschreibt basierend auf diesen Modellen Algorithmen zur automatisierten Analyse der Modelle, um Ausprägungen einer Teilmenge der in Kapitel 3 ermittelten relevanten Eigenschaften zu bestimmen. Basierend auf zwei in Kapitel 11 dargelegten Fallstudien findet in Kapitel 12 eine Bewertung des Ansatzes statt. Zudem wird die Arbeit zusammengefasst, und es wird aus dieser Arbeit resultierender zukünftiger Handlungsbedarf aufgezeigt.

## 2 Eingrenzung der betrachteten Systeme und vorausgehende Untersuchungen

Dieses Kapitel klärt die Ausgangslage der vorliegenden Arbeit. Zunächst werden die Produktionssysteme als Untersuchungsgegenstand der Arbeit vorgestellt (Abschnitt 2.1). Hierzu wird geklärt, was eine Produktion ist und wie ein Produktionssystem generell beschrieben werden kann. Zur späteren Ableitung konkreter Methoden werden die betrachteten Systeme weiter eingegrenzt und einige Annahmen an die Systeme getroffen. Anschließend wird kurz erläutert, welches Vorgehen zur Durchführung von Änderungen an Produktionssystemen empfohlen wird (Abschnitt 2.2). Abschließend wird beschrieben, warum und wie sich Produktionssysteme im Laufe ihrer Lebensdauer ändern sowie welche Artefakte davon betroffen sind (Abschnitt 2.3).

### 2.1 Eingrenzung und Beschreibung der betrachteten Systeme

Gegenstand dieser Arbeit sind Produktionssysteme. Diese unterscheiden sich in vielerlei Hinsicht, und es ist keine einheitliche Theorie über alle Arten von Produktionssystemen vorhanden. Allgemein dienen daher Klassifizierungssysteme dazu, Gruppen von Produktionssystemen ähnlichen Verhaltens sowie ähnlicher (relevanter) Eigenschaften zu identifizieren, die mit gleichen Beschreibungsmitteln und Methoden behandelt werden können [McC95].

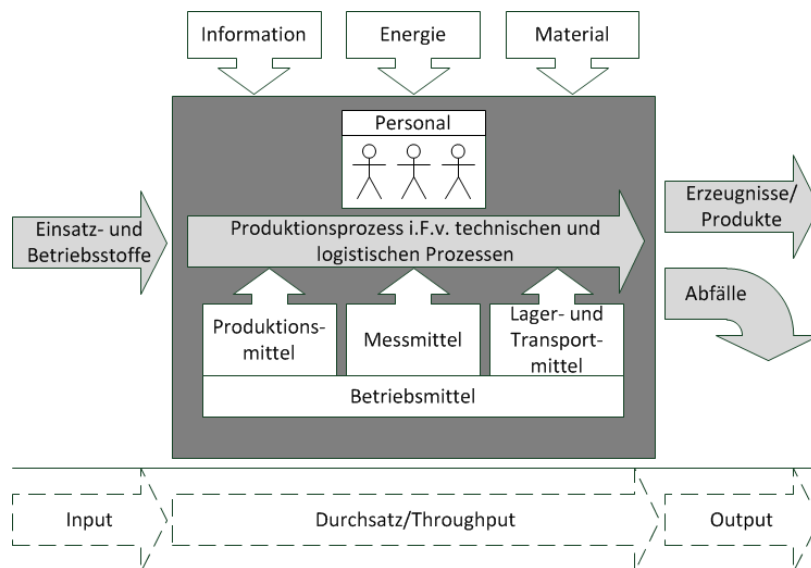
Da auch für den hier vorzustellenden Ansatz Beschreibungsmittel und Methoden benötigt werden, wird im Folgenden eine Eingrenzung der betrachteten Systeme auf Basis einer geeigneten Klassifizierung vorgenommen. Hierzu wird zunächst erläutert, was unter *Produktion* allgemein zu verstehen ist (Abschnitt 2.1.1) und die Produktionsanlage wird als *mechatronisches System* beschrieben (Abschnitt 2.1.2). Im Anschluss wird eine Unterscheidung in verfahrenstechnische und fertigungstechnische Produktion vorgenommen (Abschnitt 2.1.3). Diese dient, unter Hinzunahme einiger Annahmen, als Grundlage für eine finale Eingrenzung in Abschnitt 2.1.4. Zum Abschluss dieses Abschnitts findet eine formale Beschreibung der betrachteten Systeme statt.

#### 2.1.1 Zum Begriff der Produktion

Allgemein kann die Produktion als Prozess des Produzierens aufgefasst werden [AIK+08]. Die [DIN EN 61512-1] definiert einen (Produktions-)Prozess als „Folge von chemischen, physikalischen oder biologischen Aktivitäten für die Umwandlung, den Transport oder die Speicherung von Stoff oder Energie“. Dies deckt sich mit der in verschiedenen Forschungsbereichen verwendeten Betrachtung von Produktionssystemen als Transformationssysteme, die Materie, Information und/oder Energie von einem ursprünglichen Zustand in einen definierten Ausgangszustand bringen, vgl. [SCH99; BESÄ10; WEWA10; GRO14]. Typischerweise findet während dieses Transformationsprozesses eine Wertsteigerung bzw. Nutzenerhöhung statt und die Produktion kann, im ökonomischen Sinne, als Wertschöpfung mittels Transformation verstanden werden [Dyc06].

Häufig wird die Produktion als Input-Output-Prozess verstanden (Abbildung 2-1). Dieser Prozess generiert aus gegebenen Einsatz- und Betriebsstoffen (dem Rohmaterial) unter Zuhilfenahme von Energien, Informationen und Material mit einem gewissen Durchsatz Erzeugnisse (Produkte) und Abfälle [SCH99; SCWi04; WES06]. Um dies zu realisieren, werden Betriebsmittel (das Produktionsequipment) und Personal benötigt. Die Betriebsmittel können in Produktionsmittel (produkteigenschafts-ändernd, z.B. Werkzeugmaschinen), Messmittel sowie Lager- und Transportmittel weiter unterteilt werden. Diese Definition der Produktion umfasst auch alle betriebswirtschaftlichen Aktivitäten als Teil des Systems. Ziel der

vorliegenden Arbeit ist die Bewertung sich ändernden Systemverhaltens in Folge von technischen Änderungen. Als erste Eingrenzung soll daher im weiteren Verlauf dieser Arbeit lediglich der technische Produktionsprozess betrachtet werden. Alle nicht technischen Vorgänge werden daher nicht weiter berücksichtigt.



**Abbildung 2-1: Die Produktion als Input-Output Prozess in Anlehnung an [SCH99; ScWi04; Wes06]**

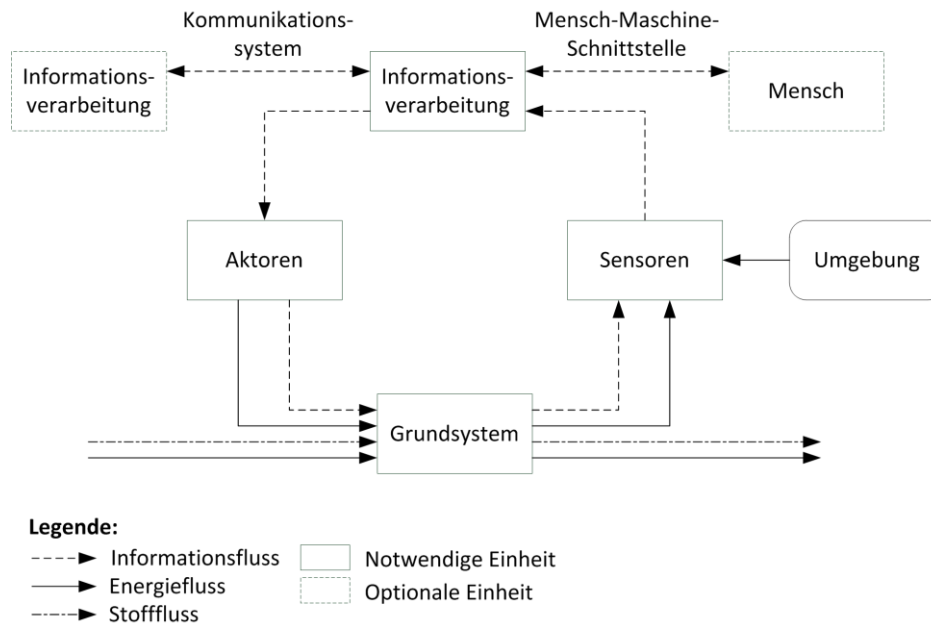
Nach Schenk [ScWi04] entspricht der technische Produktionsprozess der Kernfunktionalität der Produktion und lässt sich untergliedern in *technologische und logistische Prozesse*. Die technologischen Prozesse dienen der tatsächlichen Transformation mittels Ändern der Form und/oder Eigenschaften des Produktes mittels Produktionsoperationen und dienen der grundlegenden Wertschöpfung [ScWi04]. Produktionsoperationen lassen sich allgemein unterteilen in *Urformen, Umformen/Eigenschaftsändern, Trennen und Zusammenführen*, vgl. [CHR06; DIN 8580; Wes06; BESÄ10; GRO14]<sup>2</sup>. Weitere, nicht am direkten Wertschöpfungsprozess beteiligte Operationen sind nach [GRO14] der Transport und die Lagerung (die logistischen Prozesse), das Untersuchen und Testen sowie die Koordinierung und Steuerung. Sowohl die technologischen, als auch die logistischen Prozesse und somit der gesamte technische Produktionsprozess stehen im Fokus dieser Arbeit.

### 2.1.2 Die Produktionsanlage als mechatronisches System

Ein Produktionssystem entspricht einem i.d.R. einzigartigem mechatronischen System [Fay09; MJG11]. Nach [VDI 2206] zeichnet sich ein mechatronisches System durch „die funktionale und/oder räumliche Integration von Sensoren, Aktoren [...], Informationsverarbeitung und einem Grundsystem“ aus. Die Beziehungen zwischen diesen Elementen bestehen aus Stofffluss, Energiefluss und/oder Informationsfluss (Abbildung 2-2). Diese Definition deckt sich mit dem Verständnis einer Produktion im vorigen Abschnitt. Entsprechend der Definition eines mechatronischen Systems nach [VDI 2206] kommen in einem Produktionssystem die Disziplinen Maschinenbau, Elektrotechnik und Softwaretechnik zusammen und sein Verhalten ergibt sich aus den Interrelationen der Komponenten dieser Disziplinen [VDF+14]. Sensoren liefern Informationen über den aktuellen Anlagenzustand sowie über Produkt- und Prozesseigenschaften und Aktorsysteme erlauben die Beeinflussung des Anlagenzustandes sowie der Produkt- und Prozesseigenschaften [ASS97]. Die Informationsverarbeitung, die mittels Algorithmen in Abhängigkeit

<sup>2</sup> Eine detaillierte Auflistung von Stoff-, Energie- und Informationsoperationen findet sich in [KOKA98], ist jedoch für die weitere Betrachtung in dieser Arbeit unerheblich.

der Sensorwerte entsprechende Aktorstellgrößen vorgibt, findet im Automatisierungssystem der Anlage statt. Dieses kann in Anlehnung an das funktionale Hierarchiemodell der [DIN EN 62264-1] hierarchisch angeordnet sein (häufig wird in diesem Zusammenhang von der *Automatisierungspyramide* gesprochen). Typischerweise findet die Realisierung der unteren Automatisierungsebene mithilfe von *Speicherprogrammierbaren Steuerungen (SPS)* gemäß [DIN EN 61131-1] statt. Sie entsprechen den notwendigen Einheiten zur Informationsverarbeitung (Abbildung 2-2). In den oberen Ebenen der Automatisierungspyramide finden sich ggf. ein *Prozessleit- (PLS)* oder *SCADA<sup>3</sup>-System*, ein *Manufacturing Execution System (MES)* und ein *Enterprise Resource Planung (ERP)-System* oder ähnliche Ausprägungen wieder. Diese sind in Abbildung 2-2 als optionale Einheit zur Informationsverarbeitung zusammengefasst.



**Abbildung 2-2: Mechatronisches System nach [VDI 2206]**

Im Rahmen dieser Arbeit liegt der Fokus auf den notwendigen Einheiten eines mechatronischen Systems. Im Kontext eines automatisierten Produktionsprozesses entsprechen diese den unteren Ebenen der Automatisierungspyramide und umfassen im speziellen den Produktionsprozess sowie die Echtzeit-Automatisierung, d.h. die Steuerung mittels SPS. Die optionalen Einheiten, d.h. die Systeme auf den oberen Ebenen der Automatisierungspyramide, werden hier nicht weiter betrachtet.

### 2.1.3 Unterscheidung in diskrete Fertigung und verfahrenstechnische Produktion

Produktionssysteme weisen viele Merkmale auf, in denen sie sich unterscheiden. Eine Möglichkeit der technischen Unterscheidung ist die Unterteilung der Produktionstechnik in *Energietechnik*, *Verfahrenstechnik* und *Fertigungstechnik* [REF84; Hoi93]. Eine ähnliche und häufig verwendete Klassifizierung ist die Einteilung in verfahrenstechnische Produktion und die Fertigung diskreter Güter, vgl. [ASS97; CHR06; ScGR08; BESÄ10]. Die verfahrenstechnische Produktion beinhaltet dabei sowohl Fließ- als auch Chargenprozesse [ScGR08], bzw. kontinuierliche Prozesse und diskontinuierliche Prozesse [DIN EN 61512-1]. Zörnlein [ZÖR88] beschreibt Fertigungsprozesse als Stückprozesse, in denen sich einzeln identifizierbare Stücke in ihrer räumlichen Position und/oder in ihrem Zustand ändern. Als Ebenbild dazu beschreibt Mersch [MBS+11] als wesentliches Charakteristikum der Verfahrenstechnik die Formlosigkeit des Pro-

<sup>3</sup> Supervisory Control and Data Acquisition

duktes. Produkte der Verfahrenstechnik sind dementsprechend nicht feste Stoffe wie Gase, Flüssigkeiten oder Schüttgüter, während Produkte der Fertigung diskreten Stückgütern entsprechen.

Die Sinnhaftigkeit der Unterscheidung in verfahrenstechnische und fertigungstechnische Produktion zeigt sich auch in der einhelligen Kategorisierung der Leittechnik in Verfahrens- und Fertigungsleittechnik, vgl. [ASS97; DIN V 19222; BMQ+10]<sup>4</sup>. Auch das Engineering von verfahrenstechnischen und fertigungstechnischen Produktionssystemen und die dabei verwendeten Dokumente unterscheiden sich wesentlich [AGM+90; Eck15]. Die stark getrennte Betrachtung zwischen verfahrenstechnischer Industrie und Fertigungsindustrie zeigt sich des Weiteren darin, dass für Beide eigene Gremien, Tagungen, Zeitschriften, Normen, Modelle und Begriffssysteme sowie Werkzeuge und Systeme existieren [MBS+11]. Dies konnten auch bisherige Bemühungen, die verwendeten (leittechnischen) Begriffe [AGM+90] oder auch die verwendeten Modelle und Modellierungssprachen [BMQ+10] zu vereinheitlichen, nicht ändern.

Die Unterscheidung in Verfahrens- und Fertigungstechnik differenziert in erster Linie nach den im Prozess auftretenden Variablen. In der diskreten Fertigung sind Sensor- und Aktorwerte meist binär, während in der verfahrenstechnischen Industrie vermehrt kontinuierliche Variablen auftreten [GRO14]. Entsprechend sind auch die in Automatisierungssystemen diskreter Fertigung auftretenden Variablen überwiegend binärer Natur [LIFR99; SCH99].

Des Weiteren sind die beiden Industrien derzeit unterschiedlichen Rahmenbedingungen ausgesetzt. Fertigungstechnische Produktionssysteme zeichnen sich, eher als verfahrenstechnische, durch hohe Anforderungen an kurze Produktlebenszyklen, hohe Produktvarianz und somit einen Bedarf an hoher Wandlungsfähigkeit aus [BMQ+10]. Es wird zwar erwartet, dass dies in Zukunft auch in der verfahrenstechnischen Produktion an Relevanz zunehmen wird [HOF+14], heutzutage unterliegen jedoch Fertigungsanlagen häufiger Änderungen, insbesondere durch geänderte Anforderungen an den Durchsatz und das Produktspektrum [BMQ+10]. Allgemein stehen den Anforderungen an eine starke Dokumentationsstruktur in der Verfahrensindustrie die Anforderungen an hohe Flexibilität in der Fertigungsindustrie gegenüber [MBS+11]. Somit kann erwartet werden, dass Fertigungsanlagen eher undokumentierten evolutionären Änderungen unterliegen als verfahrenstechnische Produktionssysteme. Da der in dieser Arbeit vorzustellende Ansatz auf die Unterstützung dieser Art von Änderungen abzielt, werden im weiteren Verlauf dieser Arbeit Fertigungsanlagen betrachtet. Einige Aussagen der Arbeit gelten jedoch auch für verfahrenstechnische Anlagen. Kann eine Aussage sowohl auf die fertigungstechnische als auch auf die verfahrenstechnische Produktion bezogen werden, werden die Systeme allgemein als *Produktionssysteme* bezeichnet.

#### 2.1.4 Weitere Eingrenzung und Annahmen

Eine Fertigungsanlage kann im technischen Sinne allgemein als Bearbeitungssystem (Arbeitsplätze und Maschinen) und Materialsystem (Lager- und Transportsysteme) beschrieben werden [ZÖR88; WES06], die unter Zuhilfenahme eines Informationssystems die Aufgabe der Fertigung erfüllt [REF90]. Diese Definition soll auch im Rahmen dieser Arbeit gelten.

Im Speziellen wird für die betrachteten Systeme angenommen, dass sie aus mehreren durch ein Materialflusssystem verbundene Maschinen bestehen. Diese Maschinen können eine bis mehrere Fertigungs- und/oder Montageoperationen durchführen. Die Fertigung/Montage von Produkten kann durch eine oder mehrere alternative *Arbeitsfolgen* vorgenommen werden [WES06]. Im Rahmen dieser Arbeit wird

---

<sup>4</sup> Im Bereich der Leittechnik wird auch häufig von Fertigungs- und Prozessleittechnik gesprochen. Prozessleittechnik ist daher als Synonym für Verfahrensleittechnik zu verstehen.

angenommen, dass das System mithilfe einer oder mehrerer Speicherprogrammierbarer Steuerungen vollautomatisiert ist, d.h. der technische Prozess ohne menschliches Eingreifen erfolgt, vgl. [VDF+13]. Respektive werden alle Transport- und Bearbeitungsvorgänge ohne Eingriff des Menschen vorgenommen, sensorisch erfasst und mittels Aktorik gesteuert. Weitere Automatisierungsebenen, wie PLS, MES oder ERP, werden hier nicht weiter berücksichtigt, um auch Systeme einschließen zu können, die lediglich mittels SPS automatisiert sind (siehe Abschnitt 2.1.2).

Des Weiteren wird angenommen, dass die betrachteten Systeme als *Ereignisdiskrete Systeme* modelliert werden können. Der diskrete Zustand der Systeme kann als Menge binärer Sensor- und Aktorsignale beschrieben werden. Diese entsprechen den Ein- und Ausgangssignalen (Input/Output- bzw. I/O-Signale) der steuernden SPS(en). Hiermit ist nicht ausgeschlossen, dass das System auch kontinuierliche Variablen aufweist. Allerdings wird angenommen, dass der Fertigungsprozess und dessen Zustände in hinreichender Granularität hinsichtlich der in dieser Arbeit getroffenen Zielsetzung durch die binären Signale beschrieben werden kann. Dies gilt für eine Vielzahl von Fertigungssystemen wie Flow-Shop Systeme, automatische Transportlinien, Job-Shop Systeme, flexible Fertigungssysteme oder Montagesysteme, vgl. [SDB+83; MOGU96; CHR06; GRO07]. Es wird außerdem angenommen, dass das Verhalten des zu untersuchenden Systems deterministisch ist und sich lokal zyklisch wiederholt. Innerhalb seiner Grenzen kann ein Produktionssystem durchaus als deterministisch betrachtet werden, sofern die Steueralgorithmen deterministisch sind [ScGR08]. Allerdings sind die äußeren Einflüsse (Umgebung), hierzu zählen insbesondere Auftragseingänge und Warenabnahme, immer als stochastisch zu betrachten [ScGR08]. Zudem wird angenommen, dass das Verhalten der betrachteten Systeme vollständig sowie fehlerfrei in einem endlichen Zeitraum beobachtbar ist.

### 2.1.5 Formale Beschreibung

Im Folgenden wird eine formale Notation der betrachteten Systeme aufgestellt. Dies dient zum einen der eindeutigen Beschreibung der betrachteten Systeme. Zum anderen wird die Notation im weiteren Verlauf der Arbeit verwendet, um Systemeigenschaften, Metriken sowie die entworfenen Algorithmen zu vermitteln. Hierzu werden weitere Notationen notwendig sein<sup>5</sup>. Die folgende Beschreibung lehnt sich an der Beschreibung flexibler Fertigungssysteme an (bspw. [KUTe93; SBK06]), genügt aber auch der Beschreibung anderer Fertigungssysteme, vgl. [SBK06].

Die betrachteten Produktionssysteme beinhalten einen Satz *technischer Ressourcen*  $\mathbf{R}$ . Dieser besteht aus *Werkstück bearbeitenden Maschinen*,  $\mathbf{M} = \{M_m\}$  (bspw. Werkzeugmaschinen, Montageroboter) sowie einer Menge an *Transport- und Materialflussequipment*  $\mathbf{T} = \{T_t\}$  (bspw. Transportbänder, Drehische, Puffer, siehe Abschnitt 2.1.4). Die Fertigungsanlage ist allgemein befähigt einen oder mehrere verschiedene *Produkttypen* herzustellen. Dies geschieht durch die Bearbeitung von Werkstücken, die u.U. unterschiedlichen *Werkstücktypen*  $\mathbf{W} = \{W_w\}$  entsprechen. Die unterschiedlichen Werkstücktypen können sensorisch unterschieden werden (z.B. durch Messen von Material oder Form des Werkstücks). Zur Bearbeitung der Werkstücke werden *Fertigungsoperationen*  $\mathbf{\Omega} = \{\Omega_\omega\}$  auf diese angewandt (siehe Abschnitt 2.1.1). Maschinen können ggf. mehrere Fertigungsoperationen durchführen. Kann eine Fertigungsoperation an einer Maschine ausgeführt werden, so wird die Operation mit  $\Omega_\omega^m$  beschrieben, um zu unterscheiden an welcher Maschine die Operation ausgeführt wird. Wird zur Bearbeitung eines bestimmten Werkstücktypen eine Fertigungsoperation ausgeführt, so wird sie mit  $\Omega_\omega^w$  bezeichnet um zu unterscheiden, welcher Werkstücktyp durch sie bearbeitet wird. Zur Beschreibung einer Bearbeitung eines Werkstücktypen an einer Maschine mittels einer spezifischen Fertigungsoperation wird der Term

<sup>5</sup> Die gesamte in dieser Arbeit verwendete formale Nomenklatur ist in Anhang A zu finden.

$\Omega_{\omega}^{mw}$  verwendet. Es ist zu beachten, dass ein Produkt durch das Ausführen unterschiedlicher Reihenfolgen von Fertigungsoperationen an einem Werkstücktypen produziert werden kann. Des Weiteren kann eine Reihenfolge von Fertigungsoperationen durch unterschiedliche Reihenfolgen von Maschinen durchgeführt werden. Jede Reihenfolge an Fertigungsoperationen zur Herstellung eines Produktes wird hier als vereinfachter *Arbeitsplan* bezeichnet und die Menge an Maschinen, die dazu verwendet wird, als *Maschinenbelegung* (vgl. hierzu [SPR16<sup>6</sup>]). Produkte, die mittels desselben Arbeitsplanes hergestellt werden, werden auch dem gleichen Produkttypen zugeordnet, vgl. [CuFE10].

## 2.2 Vorgehen im Engineering von Produktionssystemen und ihrer Evolution

Wie die Realisierung eines Systems zu planen und vorzunehmen ist, wird meist anhand von Vorgehensmodellen beschrieben. Vorgehensmodelle dienen dem systematischen und standardisierten Vorgehen beim Entwurf und der Umsetzung von technischen Systemen [Eck15], indem sie die einzelnen Schritte im Engineering in Form von Aktivitäten oder Phasen beschreiben [ASS97]. Zur Durchführung von Änderungen ist die empfohlene Vorgehensweise, das gesamte Vorgehensmodell [VFS+15] oder Teile dessen [DLH11] erneut zu durchlaufen. Somit entsprechen Vorgehensmodelle zur Entwicklung eines Produktionssystems auch den empfohlenen Vorgehensmodellen zur Änderung solcher Systeme. Sie beschreiben demnach sowohl das Engineering als auch das Re-Engineering der Systeme. Unter Re-Engineering wird der Prozess zur Verbesserung eines Systems mit dem Ziel der Beseitigung von Mängeln (bzw. Degeneration) oder zur kontinuierlichen Verbesserung mittels Anpassung verstanden [ISO/IEC 26702]. Hierzu ist eine Analyse sämtlicher Engineering-Artefakte vonnöten [ISO/IEC 26702]. Somit ist eine solche Analysephase im Falle des Re-Engineerings dem Vorgehen zum initialen Engineering voranzustellen. Das *Änderungsmanagement* empfiehlt zusätzlich stets eine Analyse des Änderungsbedarfs und eine darauf folgende, möglichst formale, Beschreibung der Änderung vor der Durchführung sowie die Bewertung des Änderungserfolges und die Dokumentation der Änderung nach dessen Durchführung [JMR04].

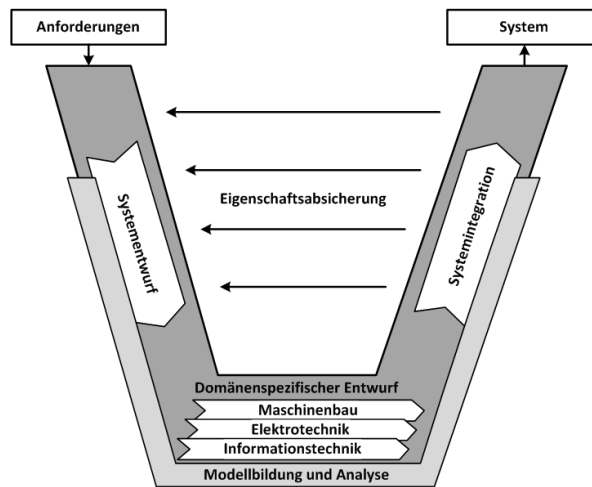
Es existiert eine Vielzahl an Vorgehensmodellen, die domänenübergreifend und domänenspezifisch in unterschiedlichen Normen standardisiert sind, vgl. [NWW97]. Die gängigen Vorgehensmodelle wie bspw. die [NA 35] zur Abwicklung von PLT<sup>6</sup>-Projekten der Automatisierungstechnik oder die Entwicklungsmethodik mechatronischer Systeme [VDI 2206] basieren dabei meist auf dem V-Modell und erweitern, konkretisieren und/oder wiederholen dieses in iterativen Schritten [Eck15]. Daher wird hier das V-Modell für mechatronische Systeme gemäß [VDI 2206] repräsentativ behandelt (Abbildung 2-3).

Der Ausgangspunkt besteht aus den Anforderungen an das System, die im Laufe des Systementwurfes konkretisiert werden. Der durchgängige Umgang mit Anforderungen während des Lebenszyklus eines Systems ist Aufgabe des *Requirements Engineering* und dem zugeordneten *Anforderungsmanagement* [PoRu11; SCH11]. Üblicherweise werden allgemeine, vom Anlagenbetreiber ausgehende, Anforderungen zu Beginn in einem *Lastenheft* festgehalten und die daraus resultierenden Anforderungen an eine technische Lösung im *Pflichtenheft* [VDI/VDE 3694]. Beide sind somit als initiale Anforderungsdokumente zu betrachten. Typischerweise umfassen sie Struktur, Funktion und Verhalten des Systems [PoRu11]. Im Pflichtenheft hat eine Formalisierung der Anforderungen meist nicht stattgefunden, womit es einer informellen Anforderungsspezifikation entspricht [SJF11b]. Eine informelle Spezifikation ist eine umgangssprachliche, an keine wohldefinierte Syntax und Semantik gebundene, Beschreibung der Aufgabenstellung, die meist in natürlicher Sprache vorliegt [VDI 2221]. Natürliche Sprache birgt jedoch die Gefahr der Mehrdeutigkeit und Subjektivität u.a. aufgrund unvollständig formulierter Anforderungen oder nicht vorhandener Begriffsdefinitionen [VDI 2221; PoRu11]. Werden Anforderungen in Form von Modellen

---

<sup>6</sup> Prozessleittechnik

ausgedrückt, wird von einem *modellgetriebenen Requirements Engineering* gesprochen [LIE07]. Die Modelle werden in einem Formalisierungsprozess gewonnen und entsprechen einer *formalen Spezifikation* der Anforderungen. Aus einer (formalen) Anforderungsspezifikation kann eine *Design- bzw. System-spezifikation* abgeleitet werden, die wiederum als Vorlage zur Realisierung des Systems dient [BEÄ10].



**Abbildung 2-3: V-Modell zur Entwicklung mechatronischer Systeme nach [VDI 2206]**

Entlang des linken Astes des V-Modells (Abbildung 2-3) werden die geforderten Eigenschaften des Systems in Top-Down-Weise in logische Teilfunktionen zerlegt, denen Teil-Spezifikationen zugeordnet werden. Basierend auf dem resultierenden Lösungskonzept wird in den einzelnen Domänen mechatronischer Systeme (Maschinenbau, Elektrotechnik, Informationstechnik) in einem letzten Konkretisierungsschritt die Detailentwicklung vorgenommen. Jeder Entwicklungsschritt durchläuft dabei seinen eigenen Entwurfszyklus auf Mikroebene. Die Systemlösungen werden entlang des rechten Astes des V-Modells (Abbildung 2-3) zunächst zu Teilsystemen und dann wiederum zum Gesamtsystem integriert. Jedem Integrationsschritt folgt eine Bewertung, in der geprüft wird, ob das/die (Teil-) System/e den formulierten Anforderungen entspricht. Der Entwicklungsprozess wird von einem Modellbildungs- und Analyseprozess begleitet [VDI 2206].

## 2.3 Evolutionäre Änderungen von Produktionssystemen

Nachdem nun die betrachteten Systeme sowie das Vorgehen zum (Re-)Engineering der Anlagen erläutert wurden, betrachtet dieser Abschnitt die Evolution der Anlagen während ihrer operativen Phase. Neben der Frage *warum* Änderungen vorgenommen werden, ist u.a. der technische Aspekt des *Wie* und *Was* von besonderer Bedeutung [LERA02; BMZ+05]. Daher steht hier im Mittelpunkt, was unter einer Evolution eines Produktionssystems zu verstehen ist (Abschnitt 2.3.1), warum diese stattfindet (2.3.2) und welche Artefakte in welcher Weise von welchen Evolutionsszenarien betroffen sind (2.3.3).

### 2.3.1 Begriffsdefinitionen

Eine technische Änderung an einem Produktionssystem kann in Anlehnung an die Änderungsmanagement-Studie von 2012 [LWM+12] wie folgt definiert werden:

**Definition 2-1: Technische Änderung in Anlehnung an [LWM+12]**

Eine technische Änderung ist eine Adaption an Teilen, Dokumenten oder Software, die bereits während des Entwicklungsprozesses freigegeben wurden.

Diese Definition schließt alle Maßnahmen zur Weiterentwicklung sowie zur Reduktion der Degradation eines Systems ein. Änderungen, die einer technischen Alterung entgegenwirken und dazu dienen, das System in Betrieb zu halten (lebenserhaltende Maßnahmen), können als *Instandhaltung* bzw. *Wartung* zusammengefasst werden. Unter Hinzunahme aller Änderungen, die nicht von vornherein geplant wurden und eher einer Anpassung an geänderte Rahmenbedingungen<sup>7</sup> entsprechen, kann unter Betrachtung der gesamten operativen Phase des Systems von einer *System-Evolution* gesprochen werden [GOG08]. Rowe et al. [RLL98] definieren den Begriff *Evolvierbarkeit* eines technischen Systems wie folgt:

„[...] the ability of a system to adapt in response to changes in its environment, requirements and implementation technologies“.

In Anlehnung hieran und ergänzt um die technische Alterung und die Instandhaltung, wird der Begriff *Evolution* im Kontext von Produktionssystemen für diese Arbeit wie folgt definiert:

**Definition 2-2: Evolution eines Produktionssystems**

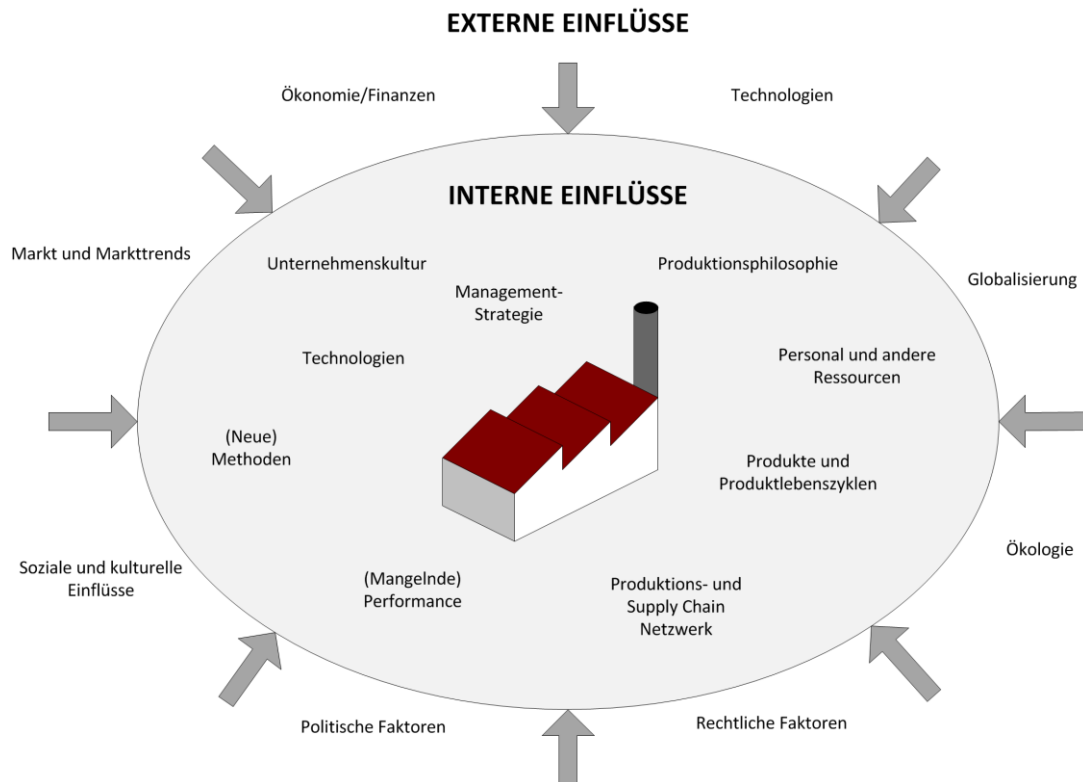
Evolution schließt sämtliche Änderungen am Produktionssystem während seiner Betriebsphase ein. Sie umfasst neben Alterungserscheinungen alle technischen Änderungen aufgrund von Wartungstätigkeiten und Anpassungen des Systems an geänderte Umwelteinflüsse, Anforderungen oder Technologien.

Besonders relevant ist hierbei die verhaltensändernde Evolution, d.h. alle technischen Änderungen, die sich auf das Verhalten des Systems auswirken, denn diese beeinflussen auch dessen Eigenschaften [HWB+13<sup>#</sup>].

### 2.3.2 Evolutionstreiber eines Produktionssystems

Produktionssysteme haben eine hohe Lebensdauer, die je nach Branche von 7 bis 50 Jahren reichen kann [ZVE10]. Ähnlich wie dies für Software gilt [LEH80], sind die Betriebskosten einer solchen Anlage höher als ihre Anschaffungskosten [SCH10]. Um ein Gewinn-Optimum zu erreichen, ist ein Produktionssystem entsprechend so lange in Betrieb zu halten, wie es ausreichend Profit abwirft, jedoch auch nicht kürzer oder länger [ROG11]. Neben der technischen Alterung und der Instandhaltung, die eine eher ungewollte Evolution hervorrufen, stehen technische Systeme einer Vielzahl von Einflussfaktoren gegenüber, die einen Änderungsbedarf hervorrufen können. Häufig steht die Produktion dabei auch in Wechselwirkung mit ihren Einflussfaktoren [Dyc06]. Die Faktoren sind bedingt durch Politik, Markt, Standards und technische Vorgaben, Kultur, Unternehmens-Organisation sowie die Anlagen-Physik [IEEE STD 1233]. Häufig wirken sich die Einflussfaktoren bereits in der Entwicklungsphase auf die Produktion aus [BeSÄ10]. Diese Einflussfaktoren, die ebenfalls während der Betriebsphase auf die Produktion einwirken, werden typischerweise in *interne* und *externe Einflüsse* unterschieden. Diese sind in Abbildung 2-4 gemäß [ScWi04; Dyc06; WEN+07; AIK+08; NHR+08; SCH08; ELWi09; BeSÄ10; ROG11] zusammenfassend dargestellt.

<sup>7</sup> Diese schließen ebenfalls geänderte Anforderungen und somit eine Weiterentwicklung ein

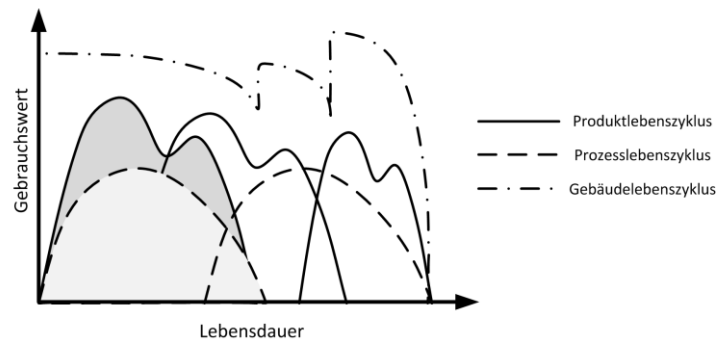


**Abbildung 2-4: Interne und externe Einflüsse auf die Produktion**

Viele der Einflussfaktoren wirken dabei zunächst auf einzelne Produkte oder das Produktportfolio [OPH05] und führen dazu, dass die Produktionsmenge, das Produktportfolio oder die Produktqualität angepasst werden müssen. Immer kürzere Lebenszyklen der Produkte erhöhen den Bedarf an Berücksichtigung der Langlebigkeit von Produktionssystemen, da es ökonomisch nicht zu rechtfertigen ist, eine Anlage lediglich für ein stark limitiertes Produktportfolio zu errichten [BESÄ10]. Folglich evolvieren die Produkte oder Produktfamilien über die Zeit und die Anlagen müssen entsprechend befähigt werden diese zu produzieren [ELM09]. Infolgedessen werden bestehende Produktionssysteme adaptiert und deren Verhalten angepasst [LBK+12]. Dieser Effekt erzeugt eine Co-Evolution von Produkt und Produktionssystem [ALEL11]. Die Zyklen dieser Co-Evolution werden bei dem zu erwartenden Anstieg an Kundenindividualisierung und entsprechendem Abfall an Losgrößen immer kürzer, vgl. [DAS03; ELM09; ALEL11; DLH11; VDMc14]. Die Lebenszyklen von Produktionsmaschinen und –anlagen sind somit unweigerlich länger als die der Produkte, die sie produzieren [ELM09; ELW109]. Insbesondere diese unterschiedlichen Lebensläufe von Produkten und Prozessen beeinflussen die Anlage. Anpassungen sind unvermeidbar, was zu dem in Abbildung 2-5 vereinfacht dargestellten Verhältnissen zwischen Produkt- und Prozesslebenszyklus im Laufe eines Anlagen-Lebenszyklus führt, vgl. [SCW104; WEN+07; ELW109].

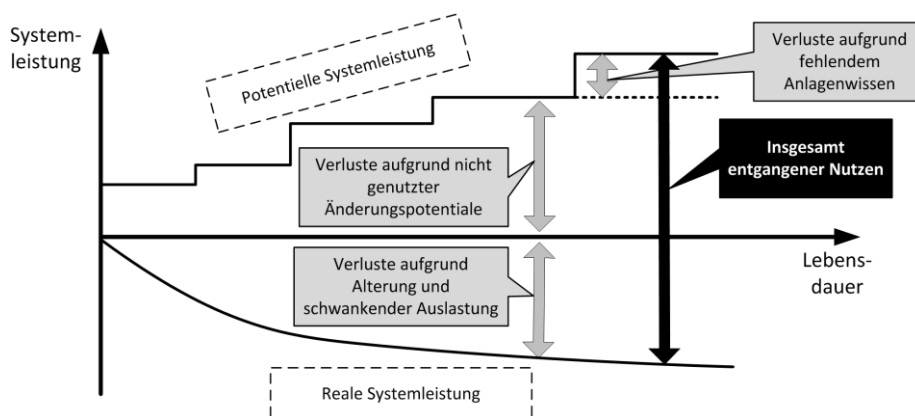
Werden die Adaptionen der Anlage nicht durchgeführt, so degeneriert sie. Bereits in den 70er Jahren stellte Lehman die Theorie auf, dass Softwaresysteme, sofern sie nicht entsprechend angepasst werden, in einer sich ändernden Umgebung altern und relativ zu Ihrer Umgebung degenerieren [LEH78]. Diese These konnte mehrfach bestätigt werden, vgl. [LEH80; LRW+97; LERA02] und gilt auch für mechatronische Systeme wie Produktionssysteme, vgl. [VFS+15]. Die Qualität der Anlagen wird daher weniger durch Verschleiß als durch technische Überalterung limitiert sofern sie nicht entsprechend adaptiert werden [NIE09]. Zudem wird die Abschätzung des Bedarfes durch die Absatzmärkte, respektive die benötigte Produktionskapazität, immer schwieriger und folglich besteht über zukünftige Sachverhalte meist Unsicherheit [AIK+08]. Je größer der Planungshorizont einer Anlage, umso größer sind die Unsi-

cherheiten über die spätere Bestimmung [MIMÖ08]. Existiert eine Differenz zwischen den Annahmen zur späteren Bestimmung des Systems und dem tatsächlich auftretenden Verwendungszweck, kann das Systemdesign die späteren Real-Anforderungen an das System nicht optimal erfüllen. Dies stellt eine weitere Art der Degradation dar. Ein zusätzlicher Grund für Anlagendegradation ist fehlendes Anlagen Know-how aufgrund unzulänglicher oder fehlender Dokumentation [NIÉ09; SDK+11]. Dies liegt darin begründet, dass eine optimale Nutzung des Systems nur möglich ist, wenn das Wissen über das System explizit verfügbar ist. Der beschriebene Degradations-Vorgang im Laufe der Betriebsphase eines Produktionssystems ist qualitativ in Abbildung 2-6 dargestellt.



**Abbildung 2-5: Verhältnis zwischen Produkt- und Prozesslebenszyklus nach [ScWi04; WEN+07; ELWi09]**

Hinzu kommt die Ungewissheit über das technisch bedingte Verhalten der Anlage. Bereits bei der Planung ist es i.d.R. nicht möglich, sämtliche technischen Sachverhalte und Zusammenhänge zu ermitteln und zu berücksichtigen [AIK+08]. Daher ist es gängig, dass das endgültige Produktionssystem schon im Zuge der Inbetriebnahme ersten Änderungen ausgesetzt ist [BESÄ10]. Aus diesen Gründen werden Produktionssysteme während ihres gesamten Lebenszyklus immer häufiger adaptiert und die Nutzungsphasen ohne Änderung werden immer kürzer [ELM09; MIMÖ08; ROG11; SCH10]. Insbesondere in der Fertigungstechnik werden teilweise, je nach Branche und Geschäftsmodell, Anlagenkonfigurationen lediglich einige Tage oder Wochen genutzt [BMQ+10]. Auch Lehman erkannte früh, dass Systeme in Benutzung häufig Änderungen unterzogen werden, was unweigerlich zu steigender Komplexität der Systeme führt [LRW+97].



**Abbildung 2-6: Degradation eines Produktionssystems während der Betriebsphase nach [NIÉ09]**

### 2.3.3 Unterscheidung von Evolutionsszenarien

Die Änderungstreiber eines Produktionssystems wirken sich üblicherweise auf die an die Anlage gestellten Anforderungen aus [RLL98]. Insbesondere funktionale Evolution von Produktionssystemen ist meist

bedingt durch geänderte Anforderungen [ELM05]. Diese Anforderungen können explizit oder implizit vorhanden sein [LHW+13<sup>#</sup>; VFS+15] und resultieren meist in Forderungen an die Fähigkeiten eines Systems [LFV13]. Außerdem können Systemanpassungen zur Erhaltung von Anforderungen notwendig sein (interne, systemimmanente Änderungen) [HAU06]. Diese Änderungen sind i.d.R. unvorhergesehen, meist sehr kurzfristig zu erledigen und daher kritisch zu betrachten [WAL12].

Änderungen an Produktionssystemen reichen von kleineren häufig durchgeführten Anpassungen bis hin zu aufwändigen Um- und Neustrukturierungen. Hinzu kommt, dass produzierende Unternehmen auf unterschiedliche Weisen auf Änderungen der externen und internen Einflussfaktoren reagieren können [NHR+08]. Hieraus ergibt sich eine Vielzahl von möglichen Systemadaptionen.

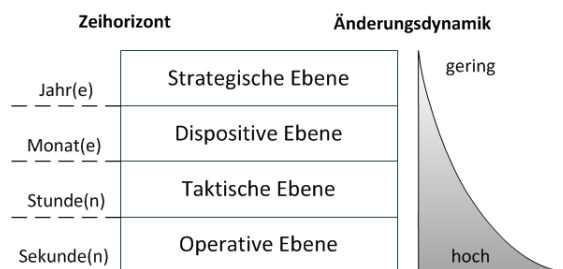
Sämtliche Änderungen können entweder *geplant* oder *ungeplant* sein; je aufwändiger eine Änderung ist, umso wahrscheinlicher ist jedoch auch, dass diese zumindest mittelfristig geplant ist [BESÄ10]. Eine Kategorisierung nach Bellgran und Säfsten [BESÄ10] unterscheidet daher zunächst grob, ob eine Änderung eher geringfügig oder tiefgreifend ist und ob sie extern oder intern getrieben ist (Abbildung 2-7).

		Änderungstreiber	
		Intern	Extern
Änderungsgrad	Geringfügig	Verbesserung	Entwicklung
	Tiefgreifend	Transformation	Revolution

**Abbildung 2-7: Arten von Änderungen in Abhängigkeit des Änderungstreibers und des Änderungsgrades nach [BESÄ10]**

Die am häufigsten vorgenommene Änderung ist hierbei die (kontinuierliche) *Verbesserung*, die als intern getrieben und geringfügig im Umfang eingestuft wird. Ihr liegt das bereits beschriebene Verständnis zu Grunde, dass jedes System ab Zeitpunkt seiner Errichtung einem Effizienzverlust bzw. einer Alterung unterliegt, sofern es nicht permanent angepasst wird [Wes06]. Insbesondere kontinuierliche Verbesserungsprozesse beginnen daher bereits mit Beginn des Regelbetriebes einer Anlage [SCH10]. Verbesserungsvorhaben sind der Instandhaltung zuzuordnen [DIN 31051]. Auch Wartung und Instandsetzung sind den geringfügigen, intern getriebenen Änderungen zuzuordnen, falls sie eine Verhaltensänderung der Anlage zur Folge haben. Sind geringfügige Änderungen extern getrieben, so entspricht dies einer *Entwicklung*. Die tiefgreifenden Änderungen können nach *Transformation* (intern getrieben) und *Revolution* (extern getrieben) unterschieden werden. Wichtig ist, insbesondere für die tiefgreifenden Änderungen, dass alle vorangegangenen Änderungen bekannt und dokumentiert sind, um einen effizienten Änderungsprozess durchführen zu können, vgl. [ALE11].

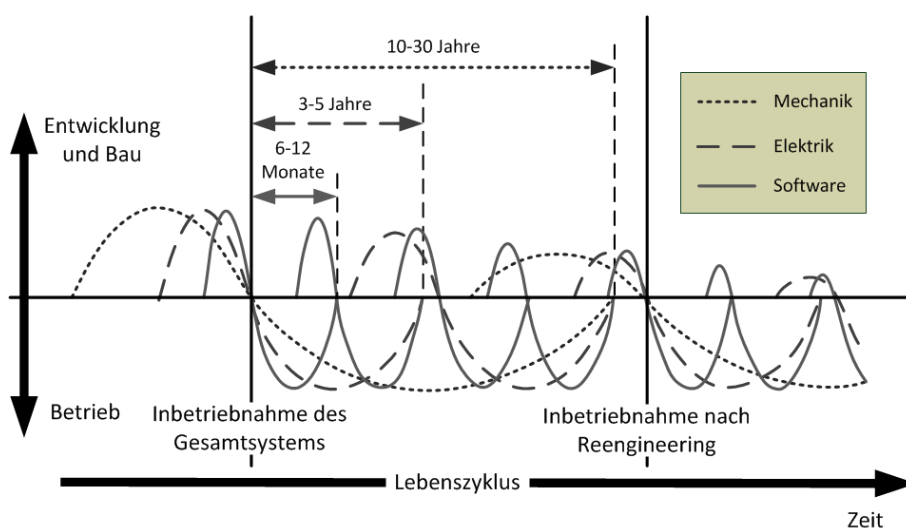
Des Weiteren können Änderungen anhand ihrer Zeitdimension unterschieden werden, z.B. entlang des 4-Schichtenmodells nach Schnieder [SCH99], wie in [SJF11A] vorgeschlagen (Abbildung 2-8). Insbesondere die *operative Ebene* (eigentliche Ausführung der Produktion) und die *taktische Ebene* (Koordinierung der Ressourcen bzw. Planung für die operative Ebene) unterliegen vielen Änderungen (vgl. auch kontinuierliche Verbesserung oben). Die Änderungsdynamik bzw. der Zeithorizont korreliert dabei meist mit dem Änderungsgrad, d.h. häufige, kurzfristige Änderungen sind meist eher geringfügig, während geplante, tiefgreifende Änderungen seltener sind und deren Durchführung länger dauert, vgl. [VFS+15].



**Abbildung 2-8: Änderungsdynamik entlang des Zeithorizontes anhand der Entscheidungshierarchie nach [SJF11A]**

Wie in Abschnitt 2.1.2 erläutert, kommen in einem Produktionssystem die Disziplinen Maschinenbau, Elektrotechnik und Softwaretechnik zusammen. Änderungen können entsprechend an der (Automatisierungs-)Software, Mechanik und/oder Elektrik vorgenommen werden [BBO+12; FeVo13], um die geforderten Eigenschaften zu implementieren. Gewünschte Verhaltensänderungen können zudem nicht nur durch eine einzige Maßnahme erreicht werden. Vielmehr stehen häufig verschiedene Aktionen in den unterschiedlichen Disziplinen zur Erreichung des Ziels zur Verfügung, vgl. [JFW+11]. Betriebsmittel einer Anlage (siehe Abschnitt 2.1.2) und ihr Verhalten sowie deren physikalischen und durch Automatisierungssoftware bedingten logischen Interrelationen können zur Erfüllung der neuen oder geänderten Anforderungen *ergänzt*, *reduziert* oder *modifiziert* werden, vgl. [BMZ+05; ELM09]. Dies führt zu einer Vielzahl an möglichen Änderungen, die üblicherweise in Verhaltensänderungen des Gesamtsystems resultieren (siehe auch [LHF+13<sup>#</sup>])<sup>8</sup>.

Die häufigsten Änderungen werden an der Software vorgenommen [VDF+13], da diese am flexibelsten ist und Adaptionen entsprechend mit wenig Aufwand zu implementieren sind. Elektrik unterliegt schneller technischer Überholung und Alterserscheinungen als Mechanik, weshalb Änderungen an der Elektrik auch häufiger vorgenommen werden. Daraus resultiert eine Azyklichkeit der Änderungen zwischen den unterschiedlichen Disziplinen [LBK+12; VDF+13] (Abbildung 2-9).



**Abbildung 2-9: Änderungszyklen der unterschiedlichen Disziplinen nach [VDF+13]**

<sup>8</sup> Detaillierter betrachtet können insbesondere physikalische Objekte auch ausfallen, gestört und wiederhergestellt werden [GÖFA12A]. Allerdings können diese Zustandsübergänge abstrahiert weiterhin den obigen Änderungsszenarien zugeordnet werden.

Evolution ist jedoch auch häufig interdisziplinär und Änderungen in Komponenten einer Disziplin können Änderungen in anderen Disziplinen erfordern [LBK+12; LHW+13<sup>#</sup>; VFS+15]. Beispielsweise ziehen Änderungen der Physik eines Produktionssystems häufig auch Änderungen der Software nach sich [LBK+12; VFS+15]. Im Falle von adaptierter Automatisierungshardware ist eine Anpassung der Software meist unumgänglich, da die Software auf die geänderte Hardwarekonfiguration angepasst werden muss [NIG14].

Aufgrund der Interdisziplinarität der Systeme, der fehlenden strukturellen Deckungsgleichheit zwischen den Disziplinen [BBO+12; GöFA12B], den teilweise nicht expliziten Abhängigkeiten zwischen den Disziplinen [JFW+11] und der u.a. daraus resultierenden Komplexität der Systeme [VDF+13] ist es wahrscheinlich, dass durchgeführte Änderungen zu unvorhergesehenen und ungewollten Verhaltensänderungen führen [KOL09; JFW+11; WIT12; FKV14; USV+14]. Des Weiteren können weitere Randbedingungen, wie bspw. ein nicht baugleiches Ersatz-Bauteil im Falle eines Ausfalls, zu ungewollten Verhaltensänderungen führen [VRM+15]. Insbesondere die nicht-funktionalen Eigenschaften werden von sämtlichen Änderungen, häufig unbemerkt, tangiert [VFS+15]. Dieser Effekt wird auch als kombinatorischer Effekt bezeichnet [VAMA12]. Meist kann das Verhalten und der Zustand eines Produktionssystems somit nicht als Folge einzelner Ursachen erklärt werden [KOL09], da die Wirkungskette einzelner Änderungen nicht offensichtlich und schwer nachvollziehbar ist [VFS+15].

Der oben erläuterte Zusammenhang zwischen Anforderungen, den geforderten Fähigkeiten an ein System sowie den unterschiedlichen Disziplinen eines Produktionssystems im Laufe der Anlagenevolution wurde von Legat et al. [LFV13] zu einem Evolutionsmodell zusammengefasst, das sinngemäß<sup>9</sup> in Abbildung 2-10 dargestellt ist.

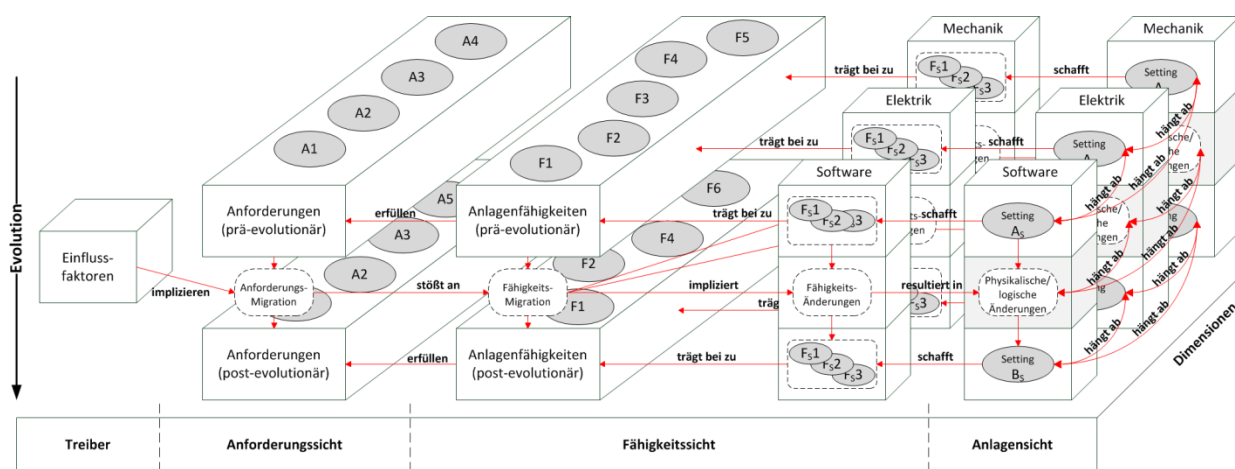


Abbildung 2-10: Evolutionsmodell nach [LFV13]

Zusammenfassend kann die Evolution die Anforderungen an das System sowie dessen Disziplinen betreffen. Anforderungen können dabei informell und/oder formal vorliegen. Aufgrund der Interrelationen der Disziplinen können zudem kausale Zusammenhänge zwischen Adaptionen dieser zur Erfüllung der Anforderungen vorherrschen. Daraus ergeben sich unterschiedliche Kategorien von Evolutionsszenarien, die erstmals in [LHW+13<sup>#</sup>] veröffentlicht wurden. Diese wurden in [VFS+15] durch die Angabe, ob die Evolutionsszenarien erwarteten Änderungen entsprechen, sowie, ob sie im laufenden Betrieb vorgenommen werden, erweitert. Die resultierenden 6 Kategorien von Evolutionsszenarien sind in Tabelle 2-1

<sup>9</sup> Legat et al. sprechen anstelle der Disziplinen Software, Elektrik und Mechanik von Software, Plattform und Kontext. Diese Betrachtungsweise ist sinngemäß die gleiche, impliziert jedoch stets einen Fokus auf die Software, weshalb hier auf diese Terminologie verzichtet wurde.

dargestellt. Die Evolutionsszenarien unterscheiden in erster Linie darin, ob und in welcher Reihenfolge (gegeben durch die Ziffern in der Tabelle) Änderungen an den informellen Anforderungen (a), der daraus abgeleiteten (semi-) formalen Spezifikation (b) sowie den Artefakten der unterschiedlichen Disziplinen (c)-(e) vorgenommen werden.

Die gewollten und längerfristig geplanten Änderungen folgen i.d.R. definierten Re-Engineering-Prozessen (vgl. Abschnitt 2.2) und es existieren zahlreiche Beschreibungsmittel, Methoden und Tools zur Unterstützung, vgl. [GÖR13]. Diese werden durch die Kategorien I und II beschrieben, die sich dadurch auszeichnen, dass die Änderungen durch geänderte Anforderungen getrieben sind und eine Anpassung der Systemspezifikation stattgefunden hat. Kategorie I sieht hierbei vor, dass lediglich eine Disziplin (a-c), oder nur die Physik (ergo Elektrik und Mechanik, d) angepasst werden muss, jedoch nicht Software und Physik gemeinsam. Häufig zieht die Änderung der Physik eine notwendige Anpassung der Software nach sich, s.o. Dieser kausale Zusammenhang ist in Kategorie II abgedeckt, die Softwareänderungen als Folge von Änderungen an der Physik vorsieht. Buckley et al. [BMZ+05] definieren eine erwartete Änderung als eine Änderung, die während des Entwicklungsprozesses eines Systems bereits Einfluss auf die Design-Entscheidungen hat. Vogel-Heuser et al. [VFS+15] erweitern diese Definition und fordern lediglich, dass ein modellgetriebener Entwicklungsprozess vorgenommen werden muss, was auch Änderungen während der Betriebsphase einschließt, die jedoch mit genügend Vorlauf vorgenommen werden können. Da in beiden Kategorien eine Änderung der Systemspezifikation und damit eines (semi-) formalen Modelles vorgenommen wird, sind die Kategorien I und II entsprechend als erwartet einzustufen. Aufgrund der Tatsache, dass die Änderungen zunächst an Modellen vorgenommen werden, sind sie des Weiteren als offline-Änderungen kategorisiert.

**Tabelle 2-1: Kategorisierung von Evolutionsszenarien nach [VFS+15]**

Kategorie		I				II			III	IV			V				VI		
		a	b	c	d	a	b	c		a	b	c	a	b	c	d	a	b	
<b>(a) Informelle Anford.</b>		1.									unverändert								
<b>(b) Systemspezifikation</b>		2.						ausgelassen											
	<b>(c) Software</b>	3.				4.			2.	3.			1.				2.		
	<b>(d) Elektrik</b>		3.		3.		3.			2.	2.		2.		1.		1.	1.	
	<b>(e) Mechanik</b>			3.		3.		3.				2.	2.			1.	1.		1.
<b>Erwartet nach [BMZ+05]</b>		ja						nein											
<b>Zeitpunkt d. Änderung</b>		offline						online											

Die Kategorien III und IV haben gemeinsam, dass sie ihren Ursprung zwar in geänderten Anforderungen haben, eine entsprechende Formalisierung dieser Anforderungen jedoch nicht stattgefunden hat. Insbesondere kurzfristige ad-hoc Änderungen in der Betriebs- oder Wartungsphase sind Teil der täglichen Arbeit [BESÄ10] und werden meist durch technisches Personal (wie bspw. Techniker) durchgeführt, die nicht die Entwickler des Systems sind [VYA13; VDF+14]. In der Regel ist das technische Betriebspersonal mit fortgeschrittenen Modellen und Methoden nicht vertraut [ScFA13]. Eine Anpassung der Spezifikation bzw. Dokumentation wird häufig unzulänglich oder gar nicht vorgenommen, da die Systemadaptionen in der operationalen Phase der Anlage und beim Betreiber vor Ort unter hohen Zeit- und Kostenrestriktio-

nen durchgeführt werden [FEVO13]. Entsprechend mangelt es an Vor- und Nachbereitung insbesondere kurzfristiger und ungewollter Änderungen [JMR04]. Kategorie III sieht vor, dass die Änderung durch Anpassung aller Disziplinen vorgenommen wird. Kategorie IV sieht Änderungen an der Mechanik und/oder Elektrik vor, was allerdings eine Anpassung der Software erfordert.

Die Kategorien V und VI sind nicht durch geänderte Anforderungen getrieben. Sie können Folge von Alterungserscheinungen und/oder Instandsetzungs- und Wartungsmaßnahmen sein, s.o. Kategorie V sieht dabei vor, dass entweder die Physik (Mechanik und/oder Elektrik) oder die Software geändert wird. Kategorie VI hingegen sieht entsprechende Software-Änderungen als Folge von Änderungen an der Physik vor.

Die Kategorien III-VI haben gemeinsam, dass eine Anpassung der formalen Spezifikation an das System nicht durchgeführt wird. Sie werden entsprechend als unerwartet und online durchgeführt eingestuft. Da die Spezifikation auch als Dokumentation des Systems dient, kann in diesem Falle auch von einer undokumentierten Evolution gesprochen werden. Eine fehlende Dokumentation erschwert nicht nur späteres Re-Engineering (vgl. Abschnitt 2.2), sondern auch eine Bewertung der Auswirkungen, die die Evolution hervorruft. Wissen über die Anlagenrealisierung wird in diesem Falle vom Anlagen-Personal gehalten und ist darüber hinaus nicht explizit verfügbar. Dies zeigt den Bedarf nach unterstützenden Ansätzen unerwarteter Evolutionsszenarien auf. Im Rahmen dieser Arbeit ist daher insbesondere die unerwartete Evolution von Interesse.

### 3 Nicht-funktionale Eigenschaften zur Bewertung evolvierender Fertigungsanlagen

Interdisziplinäre Systeme können nur als korrekt oder zufriedenstellend relativ zum Erfüllungsgrad der an sie gestellten Anforderungen bewertet werden [KMO+10]. Das vorangegangene Kapitel hat gezeigt, dass Anforderungen in einer Wechselbeziehung zur Evolution von Produktionssystemen stehen. Dies gilt sowohl für funktionale, als auch für nicht-funktionale Anforderungen (NFA) [VFS+15]. Ändert sich das Systemverhalten, ist somit ein Vergleich mit den aktuellen Anforderungen die Basis für eine Bewertung der durchgeführten Änderungen [LAN99].

Um eine generische Methode zur Bewertung der Evolution einer Klasse von Systemen zu finden, ist eine allgemeine Bewertungsgrundlage notwendig. Hier bieten sich NFAs besonders an, da sie, im Gegensatz zu funktionalen Anforderungen, einen allgemeineren Charakter besitzen und daher die Grundlage einer generelleren Bewertung bieten, vgl. [SCH11]. Die Gegenüberstellung von nicht-funktionalen Eigenschaften (NFE) und an sie gestellte Anforderungen kann insbesondere dann effizient und systemübergreifend vergleichbar vorgenommen werden, wenn die Eigenschaften in Form von objektiven quantifizierten Kennzahlen beschrieben werden [ISO/IEC 25023]. Daher soll eine Bewertung evolvierender Fertigungsanlagen im Rahmen dieser Arbeit anhand von allgemeinen und aus der Literatur ermittelten quantitativen nicht-funktionalen Eigenschaften vorgenommen werden. Demgemäß wird in diesem Abschnitt zunächst diskutiert, was unter einer NFA zu verstehen ist (Abschnitt 3.1), wie deren Erfüllungsgrad zur Bewertung des Systems allgemein ermittelt werden kann (Abschnitt 3.2) und welche NFEs und zugehörigen Metriken im Rahmen dieser Arbeit relevant sind (Abschnitt 3.3).

#### 3.1 Begriffsklärung zu nicht-funktionalen Anforderungen an Fertigungsanlagen

Allgemein entsprechen Anforderungen den expliziten oder impliziten Vorgaben und Erwartungen sämtlicher Stakeholder bzgl. der Funktionen eines Systems, in welcher Art und Weise (wie und womit) das System diese Funktionen auszuführen hat sowie in welcher Art es zu gestalten und bauen ist [SCH11]. Pohl und Rupp geben die folgende Definition des Begriffes:

**Definition 3-1: Anforderung nach [PoRu11]**

Eine Anforderung ist:

1. Eine Bedingung oder Fähigkeit, die von einem Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.
2. Eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.
3. Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß 1. oder 2.

Neben dem Artefakt, das eine Anforderung repräsentiert und dokumentiert, entsprechen Anforderungen also gewünschten Fähigkeiten und auferlegten Bedingungen an ein System. Verallgemeinert können Anforderungen gemäß 1. und 2. der obigen Definition als „Beschreibung von Eigenschaften, die das System besitzen soll“ oder „beabsichtigte Funktion“ beschrieben werden, vgl. [VDI/VDE 3542 - 2].

Im Gegensatz zum Verständnis einer funktionalen Anforderung existiert in der Literatur keine einheitliche Definition zu dem Begriff „nicht-funktionale Anforderung“ [GLI05]. Allerdings wird stets zwischen funktionalen Anforderungen bzw. Anforderungen, die die Funktion betreffen und anderen Anforderungen unterschieden [GLI07]. Pohl und Rupp, beispielsweise, untergliedern Anforderungen in *funktionale Anforderungen*, *Qualitätsanforderungen* und *Randbedingungen* [PORU11]. Die funktionalen Anforderungen werden gem. [PORU11] an das Ergebnis eines Verhaltens des Systems gestellt. Deckend dazu benennt Buede sämtliche Anforderungen an den Output eines Systems als *funktionale Anforderungen* [BUE97]. Qualitätsanforderungen und Randbedingungen können entsprechend als NFAs zusammengefasst werden [GRA11]. Jedoch werden häufig ausschließlich Qualitätsanforderungen als nicht-funktionale Anforderungen betrachtet, während Randbedingungen eine eigene Kategorie darstellen [SCH11]. Qualitätsanforderungen beziehen sich auf Qualitätsmerkmale, die nicht durch die funktionalen Anforderungen abgedeckt werden können [PORU11]. Nach Schwinn zählen die Qualitätsanforderungen zu den nicht-funktionalen Anforderungen, die sich in qualitative Anforderungen an den Systemeinsatz (z.B. bzgl. Performance, Fehlertoleranz, Portabilität, Wandlungsfähigkeit) und qualitative Sicherheitsanforderungen (z.B. bzgl. IT-Sicherheit, Verfügbarkeit) einteilen lassen [SCH11]. Randbedingungen entsprechen wiederum weiteren Einschränkungen an die Realisierung eines Systems zur Erreichung der funktionalen und der Qualitätsanforderungen [GLI07; PORU11].

Betrachtet man nun konkret Produktionssysteme, sei zunächst zu berücksichtigen, dass die Funktion der Anlagen der Änderung des Zustandes von Materie, Information und/oder Energie in einen definierten wertgesteigerten Zustand entspricht (vgl. Abschnitt 2.1). Anforderungen an diesen Zielzustand können als Anforderungen an den Produktionsprozess aufgefasst werden [BESÄ10]. Dies beinhaltet auch alle Qualitätsanforderungen an das Produkt, was sich mit dem oben erwähnten Verständnis einer funktionalen Anforderung nach [PORU11] und [BUE97] deckt, da das Produkt den Output eines Produktionssystems darstellt (vgl. 2.1.1). Analog dazu unterscheiden Rogstrand und Kjellberg zwischen Produkthanforderungen und Produktionsanforderungen<sup>10</sup> und benennen die Produkthanforderungen als funktional und die Produktionsanforderungen als nicht-funktional [ROKJ09]. Somit können funktionale Anforderungen an Produktionssysteme wie folgt definiert werden:

**Definition 3-2: Funktionale Anforderungen an Produktionssysteme**

Funktionale Anforderungen an ein Produktionssystem entsprechen der Forderung nach der Fähigkeit zur Produktion des Produkt-Portfolios und aller hierfür unerlässlich notwendigen Eigenschaften. Sie umfassen nicht nur die Schaffung der strukturellen und funktionellen Eigenschaften der Produkte, sondern auch die Erfüllung der Qualitätsanforderungen an die Produkte.

Die funktionalen Anforderungen sind meist nicht verhandelbar und unbedingt einzuhalten [BRMA97]. Gleichwohl können, bei korrekter Funktionalität, Qualitätsmerkmale wie z. B. Performance oder Wartbarkeit einen relevanten Unterschied bzgl. der Gewinnbringung eines Systems ausmachen [KRA12]. Des Weiteren sind funktionale Anforderungen spezifisch für den jeweiligen betrachteten Anwendungsfall sowie seine Struktur und erheben keinen Anspruch an die Qualität der Durchführung der zu erbringenden Leistung [SCH11]. Dies zeigt sich auch daran, dass funktionale Anforderungen im Regelfall aus den Produktspezifikationen und –rezepten gewonnen oder abgeleitet werden können [VYA13]. Auch Randbedingungen sind meist spezifisch [GLI07]. Qualitätsanforderungen sind im Gegensatz dazu häufig uni-

<sup>10</sup> Im Original “Manufacturing Requirements”

versell und projektübergreifend formulierbar, da sie sich meist auf die Art und Weise der Erfüllung der funktionalen Anforderungen beziehen [SCH11].

Aufgrund der allgemein höheren Generalität soll das Hauptaugenmerk dieser Arbeit auf den Qualitätsanforderungen an die Systeme liegen und der Begriff NFA ist im Rahmen dieser Arbeit wie folgt zu verstehen:

**Definition 3-3: Nicht-funktionale Anforderungen an Produktionssysteme**

Nicht-funktionale Anforderungen an ein Produktionssystem entsprechen allen Anforderungen, die Qualitätseigenschaften des Produktionssystems betreffen und nicht durch funktionale Anforderungen abgedeckt sind. Qualitätsanforderungen an das zu produzierende Produkt sind somit ausgeschlossen.

### 3.2 Bewertung von Produktionssystemen mittels Messung nicht-funktionaler Eigenschaften

Allgemein beschrieben umfasst eine Bewertung die Festlegung von Bewertungskriterien sowie die Messung von Eigenschaften bzw. Zielerreichungsgraden bzgl. dieser Kriterien [VDI 2206]. Die Bewertungskriterien können entsprechend als Anforderungen aufgefasst werden. Somit entspricht die Bewertung der Überprüfung des Erfüllungsgrades einer Anforderung. Die Durchführung der Bewertung geschieht mittels Analyse und häufig unter Zuhilfenahme von Modellen, die meist Teile des Systems oder bestimmte Aspekte des Systems abbilden, vgl. [LAN99]. Entsprechend gilt folgende Definition:

**Definition 3-4: Bewertung von Produktionssystemen**

Die Bewertung eines Produktionssystems umfasst den Vergleich von Eigenschaften mit festgelegten Bewertungskriterien hinsichtlich dieser Eigenschaften.

Im Folgenden werden die beiden notwendigen Schritte zur Bewertung (Eigenschaftsermittlung und Gegenüberstellung) näher diskutiert. Hierzu wird zunächst eine Modellvorstellung der Gegenüberstellung von Eigenschaften mit Bewertungskriterien (Anforderungen) aufgezeigt (Abschnitt 3.2.1). Im Anschluss wird verdeutlicht, wie die Ausprägung von Qualitätseigenschaften allgemein ermittelt werden kann (Abschnitt 3.2.2).

#### 3.2.1 Beschreibung der Bewertung anhand von Merkmalen

Wie eine Eigenschaft einer Anforderung bzw. einem Bewertungskriterium gegenübergestellt werden kann, lässt sich anhand eines Merkmalmodells beschreiben, vgl. [MEEP09; HAD15]. Die wesentlichen Elemente dieses Modells sind in Abbildung 3-1 in UML-Notation<sup>11</sup> dargestellt. Allgemein entspricht ein *Merkmal* einer klassifizierten Eigenschaft eines Objektes [EPP11]. Die durch Merkmale charakterisierten Objekte werden *Merkmalsträger* genannt [EPP11]. Entsprechend sind alle (Sub-) Systeme und Systemelemente sowie deren Funktionen, die sich durch Merkmale charakterisieren lassen, Merkmalsträger [HAD15]. Des Weiteren können auch Elementbeziehungen Merkmalsträger sein, insbesondere, wenn sie in neuen Eigenschaften resultieren [FEH+13]. Schnieder und Schnieder [ScSc10] sprechen hierbei von emergenten Eigenschaften, die die einzelne Systemelemente nicht aufweisen und ausschließlich aus der Kombination der Komponenten resultieren.

<sup>11</sup> Unified Modeling Language, zur Erläuterung der Notation siehe Anhang B

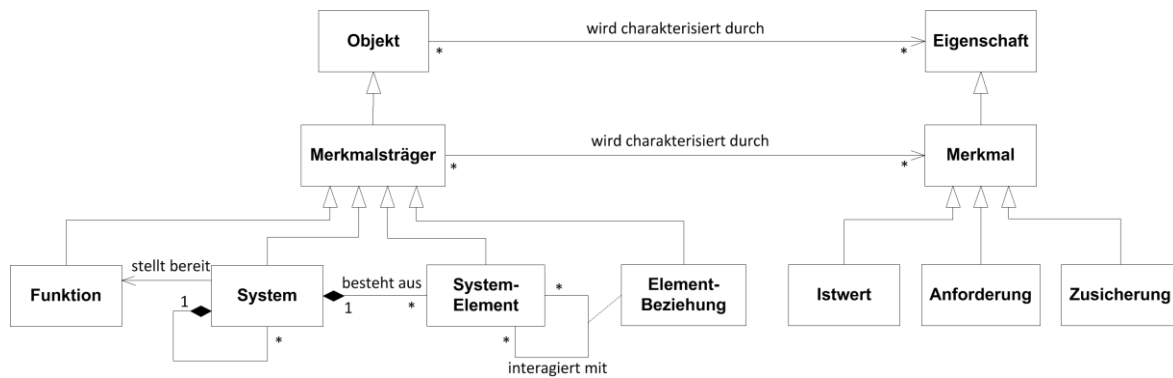


Abbildung 3-1: Merkmalmodell nach [HEE05; FEH+13; HAD15]

Der konkrete Wert, den ein Merkmal besitzt, wird als *Merkmalausprägung* bezeichnet [HEE05; EPP11]. Die Ausprägung hat dabei eine spezielle *Ausprägungsaussage* [EPP11]. Die Ausprägungsaussage wiederum besteht mindestens aus einem Aussageziel und einer Aussage [EPP11] (Abbildung 3-2). Das Aussageziel beschreibt, um was für eine Art Merkmal es sich handelt. Nach Heeg [HEE05] kann dies ein *Istwert*, eine *Zusicherung* oder eine *Anforderung* sein. Die Anforderung beschreibt eine geforderte Ausprägung. Der Istwert entspricht einer real vorhandenen Ausprägung eines Merkmals. Eine Zusicherung ist als zu erwartende Ausprägung zu verstehen und dient der Planung von Systemen<sup>12</sup>. Der Zusammenhang besteht darin, dass eine Zusicherung oder ein Istwert eine Anforderung erfüllen, falls die in der Anforderung beschriebene Aussage durch die Merkmalsausprägung des Istwertes oder der Zusicherung zutrifft [HEE05].

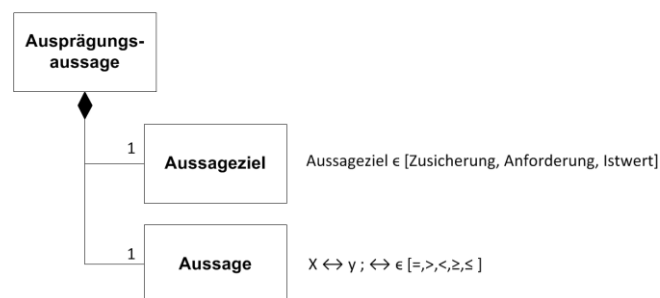


Abbildung 3-2: Ausprägungsaussage eines Merkmals nach [EPP11]

Heeg [HEE05] unterscheidet zwischen *elementaren Anforderungen* und *komplexen Anforderungen*. Komplexe Anforderungen sind durch logische Verknüpfung kombinierte elementare Anforderungen. Elementare Anforderungen entsprechen dem Vergleich einer Merkmalsausprägung in der Ausprägungsaussage mittels einfachen Grundoperatoren (gleich, größer, kleiner, größer-gleich, kleiner-gleich) mit einem festen Wert oder einer referenzierten Merkmalsausprägung. Im Falle einer Anforderung an ein Merkmal referenziert die Merkmalsausprägung mit Aussageziel „Anforderung“ also auf die Merkmalsausprägung deren Aussageziel ein „Istwert“ oder eine „Zusicherung“ ist. Eppele [EPP11] hat diese Grundoperatoren für verschieden Skalen zusammengetragen (Tabelle 3-1).

<sup>12</sup> Ein Beispiel hierfür ist eine gewisse Drehzahl oder ein Drehzahlbereich, den ein Motorenhersteller seinem Kunden zusichert.

Tabelle 3-1: Grundoperatoren der Ausprägungsaussagen nach [EPP11]

Aussageart ↔	Skala			
	Nominal	Ordinal	Intervall	Verhältnis
gleich (=)	$x = y$	$x = y$	$y - \Delta < x < y + \Delta$	$y - \Delta < x < y + \Delta$
größer (>)	-	x über y	$x > y$	$x > y$
kleiner (<)	-	x unter y	$x < y$	$x < y$
größer-gleich ( $\geq$ )	-	x über oder gleich y	$x \geq y$	$x \geq y$
kleiner-gleich ( $\leq$ )	-	x unter oder gleich y	$x \leq y$	$x \leq y$

x ist der Wert der Merkmalsausprägung, auf die sich die Aussage bezieht

y ist der Vergleichswert (Angabe als Wert oder als Referenz auf den Wert einer anderen Merkmalsausprägung)

$\Delta$  ist eine Toleranz, die im allgemeinen Merkmal fest definiert ist

Zusammenfassend entspricht die Bewertung also der Gegenüberstellung einer Istwert- oder Zusicherung-Merkmalsausprägung (Eigenschaft) mit der Anforderungsausprägung (Bewertungskriterium) mittels verschiedener Grundoperatoren (Tabelle 3-1). Im Rahmen dieser Arbeit liegt der Fokus ausschließlich auf der Gegenüberstellung des Istwertes mit der Anforderung. Es sei anzumerken, dass die Ausprägung der Anforderungsaussage system- und anwendungsfall- bzw. problemspezifisch ist. Zum Beispiel ist der geforderte Mindestdurchsatz in der Automobilfertigung ein anderer als in der Elektronikproduktion. Folglich ist die Bewertung selbst systemspezifisch vorzunehmen. Somit reduziert sich der generelle Anteil der Bewertung auf die Ermittlung der Istwert-Merkmalsausprägungen. Wie diese für Qualitätsmerkmale bzw. Qualitätseigenschaften vorzunehmen ist, wird im Folgenden diskutiert.

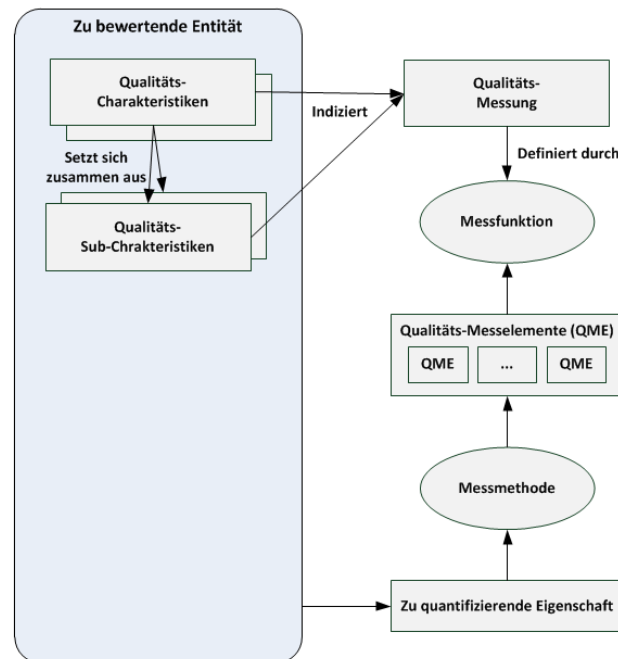
### 3.2.2 Ermittlung der Ausprägung von Qualitätsmerkmalen

Qualitätsanforderungen eignen sich aufgrund ihrer Generalität zur systemübergreifenden Bewertung (vgl. Abschnitt 3.1). Hierzu ist es notwendig, die Systemqualität, d.h. die Ausprägung der Qualitätsmerkmale, in einer geeigneten Art und Weise zu ermitteln. Die SQuaRE<sup>13</sup>-Serie der ISO/IEC beschäftigt sich mit der Definition, Spezifikation und Messung von Qualitätsanforderungen. Nach Auffassung der Standards muss Qualität über zu quantifizierende Eigenschaften gemessen werden [ISO/IEC 25021; ISO/IEC 25023]. Es wird also nach der Ermittlung von Merkmalsausprägungen mit Aussageziel „Istwert“ auf einer ordinalen Zahlenskala verlangt (vgl. Abschnitt 3.2.1). Diese werden üblicherweise in Kennzahlen ausgedrückt. Kennzahlen entsprechen einem guten Instrumentarium, da sie Grenzwerten gegenübergestellt werden können, um Anforderungen zu überprüfen oder allgemein einem Verbesserungszyklus von Produktionssystemen beizutragen [VDMA 66412-1].

Die [ISO/IEC 25023] definiert ein Referenzmodell zur Messung von Qualität (Abbildung 3-3). Im Allgemeinen setzt sich dieses aus verschiedenen *Charakteristiken* zusammen, die sich wiederum aus einer oder mehreren *Subcharakteristiken* ergeben, die evtl. weiter verschachtelt sind [ISO/IEC 25023]. Diese Charakteristiken können mittels Messung ermittelt oder geschätzt (allgemein: indiziert) werden. Der Begriff „Charakteristik“ ist hier gleich zu setzen mit dem Begriff „Merkmal“ mit Aussageziel „Istwert“ und dessen Messung entspricht der Ermittlung der Merkmalsausprägung. Die [ISO/IEC 25020] unterscheidet zwischen Basis-Messungen (base measures) und abgeleiteten Messungen (derived measures),

<sup>13</sup> Systems and software quality requirements and evaluation, umfasst die Standards ISO/IEC 2500n, 2501n, 2502n, 2503n und 2504n

die sich aus Kombinationen von Basismessungen ermitteln lassen<sup>14</sup>. Die Qualitätsmessung ist dabei durch eine Messfunktion definiert und die Basis-Messungen werden mittels einer Messmethode vorgenommen [ISO/IEC 25023].



**Abbildung 3-3: Qualitätsmessung nach [ISO/IEC 25023]**

Der Zusammenhang zwischen Messelementen, Messfunktion und Qualitätsmessung soll durch folgendes Beispiel zur Fehlerrate verdeutlicht werden (siehe Abbildung 3-4): Die Fehlerrate einer Systemkomponente lässt sich aus der Anzahl an Ausfällen in einem bestimmten Zeitraum bestimmen<sup>15</sup> [BIR99]. Um die Ausfallrate zu messen, muss daher zum einen jeder Ausfall quantitativ erfasst werden. Zum anderen muss die Laufzeit, in der die Ausfälle ermittelt wurden, gemessen werden. Diese beiden Größen entsprechen den Messelementen und deren Messung ist als Basismessung zu verstehen. Die Messfunktion beschreibt nun, wie die Fehlerrate aus den Messelementen bestimmt werden kann. Somit wird die Fehlerrate mittels abgeleiteter Messung bestimmt. Mithilfe der Fehlerrate kann auch die Zuverlässigkeit der Komponente bestimmt werden, womit die Fehlerrate wiederum eine Sub-Charakteristik der Zuverlässigkeit ist. Wird zusätzlich die Zuverlässigkeit weiterer Systemkomponenten ermittelt, so kann unter Zuhilfenahme von Strukturinformationen (die ggf. auch gemessen werden) die Zuverlässigkeit des Gesamtsystems ermittelt werden. Dies zeigt, dass durch Basismessungen ermittelte Messelemente für eine Vielzahl von Qualitätsmessungen verwendet werden können. Die Qualitätsmessungen selber unterscheiden sich lediglich durch ihre Messfunktion. Die Aufgabe der Qualitätsmessung reduziert sich damit auf die Basismessungen. Welche Basismessungen im Rahmen dieser Arbeit zur generellen Bewertung von Fertigungssystemen geeignet sind, wird im folgenden Abschnitt geklärt.

<sup>14</sup> Vgl. die Begriffe elementare Anforderung und komplexe Anforderung in Kapitel 3.2.1

<sup>15</sup> Zur detaillierten Diskussion sei auf Kapitel 3.3 verwiesen.

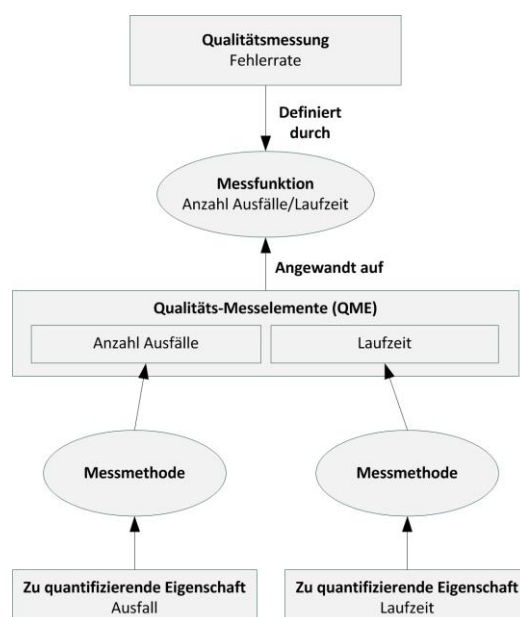


Abbildung 3-4: Beispiel zur Qualitätsmessung "Fehlerrate"

### 3.3 Kennzahlen und Metriken nicht-funktionaler Eigenschaften

Die Qualität einer Fertigung kann viele Dimensionen aufweisen und es existiert eine sehr hohe Anzahl an Qualitäts-Metriken und –Kennzahlen, vgl. [WHI96]. Dies liegt nicht zuletzt daran, dass sich verschiedenste Forschungsbereiche und Tätigkeitsfelder mit dem Thema beschäftigen. Hierzu zählen u.a. das Controlling (z.B. [BAHA09]), Benchmarking (vgl. [TAY97; SCH13]), Systems- und Production-Engineering (z.B. [NGP95; IEEE STD 1233]), Automation Engineering (z.B. [SCH99]), das Requirements Engineering (z.B. [ISO/IEC 25010]) sowie das Production Management [AWF04]. Eine vereinheitlichte und fest definierte Menge an Anforderungen an die Fertigung, wie sie bspw. für Software- und Software-intensive Systeme (siehe [ISO/IEC 25022]) sowie allgemeine Systeme aus Sicht des Systems Engineering (siehe [IEEE STD 1233]) existieren, ist nicht vorhanden [SCH13]. Daher beinhaltet dieser Abschnitt das Ergebnis einer ausführlichen Literaturrecherche zu gängigen für Fertigungssysteme relevanten Qualitätsmerkmalen und zugehörigen Metriken. Dabei liegt der Fokus auf denjenigen Qualitätsmerkmalen, die sich auf die Produktion als technischen Prozess (Produktionsperspektive, vgl. [AWF04; VDI 2893; BRO08; BAHA09; KRAR09]), auf operativer und taktischer Ebene (vgl. [LYCR91; SCH99; NYS+00; VDI 2893; VDI 4490]) mit Prozess- und Ausgangsgrößen (vgl. [NMP+00; BRO08; ROKI09]) einer Anlage und ihrer Subsysteme beziehen. Für eine Verwendung zur Bewertung von evolutionären Änderungen mittels Beobachtung des Verhaltens des Systems ist es zudem unabdingbar, dass die Eigenschaften entsprechend operationalisiert sind [VFS+15], d.h. in Kennzahlen quantifizierbar und auf Grundlage gängiger I/O-Signale (siehe Abschnitt 2.1.4) ermittelbar<sup>16</sup>. Kennzahlen bieten den Vorteil, dass sie als quantitativ erfasste Sachverhalte in konzentrierter Form zur Beurteilung von Produktion besonders geeignet sind, vgl. [ARBR72; BRMA97; LAN99; DYC06; BRA15].

Ziel dieses Abschnitts ist es, einen Satz an Messelementen (siehe Abschnitt 3.2.2) zu ermitteln, der es ermöglicht, die Ausprägungen eines möglichst großen Anteils allgemein anerkannter Qualitätsmerkmale (siehe Abschnitt 3.2.1) ermitteln oder abschätzen zu können.

<sup>16</sup> Für eine detailliertere Diskussion hierzu siehe auch [LHF+13<sup>#</sup>]

Zur Übersicht wurden die Merkmale in Performance-Merkmale (bzgl. Zeit- und Mengengrößen, Abschnitt 3.3.1), Merkmale bzgl. der Verlässlichkeit eines Systems (Abschnitt 3.3.2) sowie Flexibilitätsmerkmale (Abschnitt 3.3.3) untergliedert.<sup>17</sup>

### 3.3.1 Performance im Sinne von Zeit- und Mengen-bezogenen Leistungskennzahlen

Performance-Anforderungen werden, implizit oder explizit, an alle Systeme gestellt [BUE00]. Performance ist allerdings ein weitreichender Begriff, der in Forschung und Technik nicht klar definiert ist [SCH13]. Häufig werden sämtliche Qualitätsmerkmale unter dem Begriff Performance zusammengefasst (siehe z.B. [TAY97]). Im Rahmen dieser Arbeit wird Performance als Oberbegriff für alle zeit- und mengenbezogenen Eigenschaften eines Fertigungssystems verstanden.

Häufig wird Performance anhand von Kennzahlen ausgedrückt, die als *Key Performance Indicator (KPI)* bezeichnet werden. Zum Thema Performance-Messungen und zugehörige KPIs existieren bereits verschiedene Normen und Standards. Das VDMA Einheitenblatt [VDMA 66412-1] sowie der fast deckungsgleiche internationale Standard [ISO 22400-2] beschäftigen sich mit der Definition von KPIs zur Bewertung des Operations-Managements. Ein ähnliches Kennzahlenmodell stellt die [DIN 8743] für Verpackungsmaschinen und –Anlagen vor, was die Generalität der Kennzahlen unterstreicht. Gleiches gilt bspw. auch für die technischen Kennzahlen der VDI-Richtlinien [VDI 4400 - 2] und [VDI 3978], die KPIs und Basis-Messungen für die Logistik benennen. Weitere weitestgehend deckungsgleiche KPIs sind die Produktionskennzahlen der Arbeitsgemeinschaften für Vitale Unternehmensentwicklung e.V. [AWF04]. Aufgrund ihrer offensichtlich hohen Relevanz sollen die in den erwähnten Normen, Standards und Richtlinien definierten Kennzahlen, die in den o.g. Betrachtungshorizont fallen, im Detail vorgestellt werden. Auch wenn die diskutierten Kennzahlen in weiterer Literatur zu finden sind (vgl. z.B. [WRG+10; BRA15]), sei doch aufgrund des hohen Verbreitungsgrades in erster Linie auf die o.g. Normen, Richtlinien und Standards verwiesen. Hierzu sei erwähnt, dass sämtliche Kennzahlen, die Planungsdaten beinhalten, von der weiteren Betrachtung ausgeschlossen wurden, da Planungsdaten i.d.R. nicht in I/O-Signalen der Steuerungen reflektiert werden. Gleiches gilt für Daten bzgl. manuell oder durch Planungssysteme vorzunehmender Vorgänge wie bspw. die Belegung von Ressourcen oder der manuelle Werkzeugwechsel an einer Werkzeugmaschine (WZM). Die Daten werden nur berücksichtigt, wenn sie anderweitig approximiert werden können. Des Weiteren sei zu beachten, dass die Definitionen lediglich inhaltlich übernommen wurden und an die in dieser Arbeit vorherrschende Notation<sup>18</sup> und dessen Vokabular dargestellt sind.

#### 3.3.1.1 Beleggrad

Der Beleggrad beschreibt das Verhältnis der Zeit, die eine technische Ressource durch einen Auftrag belegt ist, zur Gesamtproduktionsdauer. Häufig wird eine solche Belegung durch ein Planungssystem vorgenommen (vgl. z.B. [KRAR09]). Da Planungsdaten hier nicht verwendet werden, wird die Belegungszeit durch die Zeit approximiert, in der eine technische Ressource tatsächlich mit einem Auftrag beschäftigt ist, d.h. nicht in ihrem Ruhezustand verharrt:

---

<sup>17</sup> Zusammenfassend werden die zur Ermittlung der Merkmalsausprägungen notwendigen Basis-Messungen in Anhang C dargestellt.

<sup>18</sup> Zur hier verwendeten Notation, siehe Anhang A

$$BLG_r = \frac{t_r^B}{t_p} \quad (3-1)$$

Mit:

$t_r^B$ : Belegungszeit der technischen Ressource  $r$ , approximiert durch die Zeit, die die technische Ressource nicht im Ruhezustand ist

$t_p$ : Gesamtproduktionsdauer

Da der Beleggrad häufig nur für Werkstück bearbeitende Maschinen relevant ist, kann der Index  $r$  (Ressource allgemein) auch durch  $m$  (Werkstück bearbeitende Maschine) ersetzt werden. Der Beleggrad, wie er hier beschrieben ist, aggregiert über alle Produktionsaufträge. Es ist denkbar, dass auch der Beleggrad pro Stück-Produktion<sup>19</sup> oder für einen Produkttypen ermittelt werden soll. Hierzu müssen die (mittlere) Produktionsdauer je Produkt (eines Produkttypen, resp. Durchlaufzeit) sowie die zugehörigen Belegzeiten heran gezogen werden.

### 3.3.1.2 Durchsatz

Der Durchsatz gibt die Menge produzierter Güter pro Zeiteinheit an. Da hier allgemein davon ausgegangen wird, dass das Fertigungssystem mehrere Produkt- bzw. Werkstücktypen bearbeitet, wird der Durchsatz für jeden Werkstücktypen  $w$  angegeben.

$$DS_w = \frac{PM_w}{t_p} \quad (3-2)$$

Mit:

$PM_w$ : Produzierte Menge des Werkstücktypen  $w$

$t_p$ : Gesamtproduktionsdauer

Der Durchsatz ist meist auch als aggregierte Größe relevant, d.h. über alle Werkstücktypen hinweg. Hierfür sollte dennoch immer die produzierte Menge eines jeden Werkstücktypen mit angegeben werden, um den Durchsatz auch für den aktuellen Produktmix interpretieren zu können.

Alternativ zur in (3-2) angegebenen Berechnung kann nach [Gro14] als Berechnung der „production rate“ auch der Kehrwert der mittleren Produktionsdauer eines Produktes herangezogen werden:

$$PR_w = \frac{1}{t_w^\mu} \quad (3-3)$$

Mit:

$t_w^\mu$ : Mittlere Produktionsdauer für Werkstücke des Typen  $w$  (Zykluszeit)

In diesem Fall erhält die Kennzahl einen leicht veränderten Charakter, da unproduktive Zeiten (z.B. umrüsten, Wartung, Ausfall etc.) während der Gesamtproduktionsdauer in der Kennzahl nicht berücksichtigt sind.

### 3.3.1.3 Nutzgrad

Der Nutzgrad gibt an, zu welchem Anteil der Belegungszeit einer Werkstücke bearbeitenden Maschine wertschöpfende Aktivitäten stattgefunden haben. Als wertschöpfende Aktivitäten werden dabei alle

---

<sup>19</sup> Eigentlich müsste hier der Produktionsauftrag berücksichtigt werden. Da dieser jedoch einem Planungsdatum entspricht, wird er hier auf das einzeln produzierte Stück reduziert.

Vorgänge bezeichnet, die das Werkstück bearbeiten bzw. verändern. Der Transport und die Lagerung sind somit ausgeschlossen.

$$NG_m = \frac{t_m^{HN}}{t_m^B} \quad (3-4)$$

Mit:

$t_m^{HN}$ : Hauptnutzungszeit der Maschine  $m$ , entspricht der Zeit, in der die Maschine wertschöpfende Aktivitäten durchführt

$t_m^B$ : Belegungszeit der Maschine  $m$ , approximiert durch die Zeit, die die Maschine nicht im Ruhezustand ist

### 3.3.1.4 Effektivität

Die Effektivität gibt an, inwieweit eine Maschine oder die gesamte Anlage ihre Hauptnutzungszeit zur Produktion genutzt hat. Sie errechnet sich aus der Multiplikation von geplanter Produktionszeit je Produkt und produzierter Menge über alle Werkstücktypen  $w$  aufaddiert geteilt durch die Hauptnutzungszeit über alle Maschinen  $m$  aufaddiert. Da Planungszeiten hier ausgeschlossen sind, wird die geplante Produktionszeit durch die tatsächliche mittlere Produktionszeit approximiert. Möglich wäre hier auch die Verwendung der maximalen Produktionsdauer (Annahme: pessimistische Planung) oder der minimalen Produktionsdauer (Annahme: optimistische Planung). Unter Berücksichtigung mehrerer Maschinen und Produkttypen ergibt sich daraus die folgende Formel:

$$Eff = \frac{\sum_w PM_w \cdot t_w^\mu}{\sum_m t_m^{HN}} \quad (3-5)$$

Mit:

$PM_w$ : Produzierte Menge des Werkstücktypen  $w$

$t_w^\mu$ : Mittlere Produktionsdauer für Werkstücke des Typen  $w$  (Zykluszeit)

$t_m^{HN}$ : Hauptnutzungszeit der Maschine  $m$

Um die Effektivität einer einzelnen Maschine zu berechnen, muss bekannt sein, wie viele Werkstücke die jeweilige Maschine durchliefen und wie viel Zeit die Durchläufe im Mittel benötigten. Die Summe im Divisor von (3-5) wird hierbei durch die Hauptnutzungszeit der entsprechenden Maschine ersetzt und  $Eff$  mit dem entsprechenden Index versehen.  $PM_w$  und  $t_w^\mu$  dürfen in dem Fall nur für die durch die Maschine tatsächlich bearbeiteten Werkstücke aufsummiert werden.

### 3.3.1.5 Qualitätsrate

Die Qualitätsrate gibt Aufschluss über die Menge fehlerhaft produzierter Ware. Sie errechnet sich aus der Anzahl fehlerfrei produzierter (bzw. nicht aufgrund minderer Qualität aussortierter) Produkte (Gutmenge) und der gesamten Produktionsmenge. Da die Qualitätsrate für unterschiedliche Werkstücktypen stark abweichend sein kann, wird hier die Berechnung der Qualitätsrate je Werkstücktyp empfohlen.

$$QR_w = \frac{GM_w}{PM_w} \quad (3-6)$$

Mit:

$GM_w$ : Gutmenge des Werkstücktyps  $w$

$PM_w$ : Produzierte Menge des Werkstücktypen  $w$

Durch Bildung eines gewichteten Mittelwertes der Qualitätsraten aller Werkstücktypen  $w$  kann die Gesamt-Qualitätsrate bestimmt werden.

### 3.3.1.6 Gesamtanlageneffektivität

Die Gesamtanlageneffektivität ( $OEE^{20}$ ) ist eine weit verbreitete Performance-Messung und entspricht einer aggregierten Größe aus weiteren Qualitätsmessungen. Sie erlaubt Aussagen über die Effektivität eines Produktionssystems unter Berücksichtigung von Ausschuss und Ausfall von Anlagenkomponenten.

$$OEE = A \cdot QR \cdot Eff \quad (3-7)$$

Mit:

$A$ : Verfügbarkeit (siehe Abschnitt 3.3.2.1)

$QR$ : Qualitätsrate (aggregiert über alle Werkstücktypen, s.o.)

$Eff$ : Effektivität (siehe Abschnitt 3.3.1.4)

### 3.3.1.7 Technischer Nutzgrad

Der technische Nutzgrad einer Maschine entspricht ihrem Wirkungsgrad. Er beschreibt den Verlust an Hauptnutzungszeit durch störungsbedingte Unterbrechungen.

$$TN_m = \frac{t_m^{HN}}{t_m^{HN} + t_m^{SU}} \quad (3-8)$$

Mit:

$t_m^{HN}$ : Hauptnutzungszeit der Maschine  $m$

$t_m^{SU}$ : Kumulierte Dauer der störungsbedingten Unterbrechungen an Maschine  $m$

### 3.3.1.8 Prozessgrad

Der Prozessgrad gibt den Anteil an Zeit wertschöpfender Aktivitäten zur gesamten Produktionsdauer an.

$$PG = \frac{\sum_m t_m^{HN}}{t_p} \quad (3-9)$$

Mit:

$t_m^{HN}$ : Hauptnutzungszeit der Maschine  $m$

$t_p$ : Gesamtproduktionsdauer

---

<sup>20</sup> Overall Equipment Effectiveness

Ähnlich wie bei dem Beleggrad kann der Prozessgrad auch pro Produktionsauftrag relevant sein. Hierfür sind die (mittlere) Durchlaufzeit des einzelnen Auftrages sowie die zugehörigen Hauptnutzungszeiten zu berücksichtigen.

### 3.3.1.9 First Pass Yield

Der First Pass Yield (FPY) ist eine Kennzahl, die Aussage über die „Ausbeute“ einer Produktion gibt. Ein niedriger FPY korreliert dabei mit niedrigen Fehlerkosten. Er errechnet sich aus der Anzahl an Gutteilen und geprüften Teilen. Im Gegensatz zur Gutmenge (siehe *Qualitätsrate*) sind die Gutteile diejenigen, die bereits im ersten Prüfdurchgang als qualitativ ausreichend befunden wurden. Wie bei der Qualitätsrate wird hier empfohlen, wenn möglich, den FPY für jeden Werkstücktypen zu ermitteln und ggf. zu aggregieren.

$$FPY_w = \frac{GT_w}{PT_w} \quad (3-10)$$

Mit:

$GT_w$ : Anzahl an Gutteilen des Werkstücktypen  $w$

$PT_w$ : Anzahl an geprüften Teilen des Werkstücktypen  $w$

Im Falle nicht Werkstücktypen bezogener Zahlen kann der Index  $w$  hier weggelassen werden.

### 3.3.1.10 Ausschussquote

Die Ausschussquote ist die Menge an produzierten Stücken, die aufgrund minderwertiger Qualität aussortiert wurde. Auch hier wird der Bezug auf den Produkttypen empfohlen.

$$AQ_w = \frac{AM_w}{PM_w} \quad (3-11)$$

Mit:

$AM_w$ : Ausschussmenge des Werkstücktypen  $w$

$PM_w$ : Produzierte Menge des Werkstücktypen  $w$

Durch Weglassen des Index  $w$  wird die über alle Werkstücktypen aggregierte Kennzahl dargestellt.

### 3.3.1.11 Nacharbeitsquote

Die Nacharbeitsquote gibt den Anteil an Teilen, die im Laufe der Produktion nachgearbeitet werden mussten. Sie gibt Aufschluss über die Menge zusätzlicher Aufwendungen. Auch hier wird der Bezug auf den Produkttypen empfohlen.

$$NQ_w = \frac{NM_w}{PM_w} \quad (3-12)$$

Mit:

$NM_w$ : Nacharbeitsmenge des Werkstücktypen  $w$

$PM_w$ : Produzierte Menge des Werkstücktypen  $w$

Durch Weglassen des Index  $w$  wird die über alle Werkstücktypen aggregierte Kennzahl dargestellt.

### 3.3.1.12 Kapazität

Eine Metrik für Produktionskapazität ist zwar in den o.g. Normen und Standards nicht (explizit) genannt, jedoch finden sich verschiedene Metriken in der Literatur (z.B. [BUS87; ELM91; ZAMU07; GRO14]), was die Wichtigkeit einer Kenngröße für die Produktionskapazität unterstreicht. Dabei gibt es verschiedene Auffassungen zum Begriff der (Produktions-) Kapazität. Kapazität ist bspw. in [BUS87] allgemein als die Obergrenze produzierbarer Güter pro Zeiteinheit definiert und gibt somit die maximale Ausgangsrate des Systems an. Hierzu können unterschiedlichste Kennzahlen definiert werden, wie bspw. die *geplante Kapazität* oder eine *Nominal-Kapazität*, vgl. [ELM91]. Hier wird die Ausgangskapazität betrachtet. Sie wird als maximale Anzahl produzierbarer Einheiten pro Zeit gemessen [ZAMU07]. Entsprechend kann sie für jeden Produkttypen aus der minimalen Zykluszeit zur Produktion berechnet werden:

$$C_w = \frac{1}{t_w^{min}} \quad (3-13)$$

Mit:

$t_w^{min}$ : Minimale Produktionsdauer für Werkstücke des Typen  $w$

Die Kapazität kann auch für einzelne Ressourcen relevant sein. Hierzu ist die Zeit des minimalen Produktionszyklus der Ressource heranzuziehen.

### 3.3.1.13 Auslastung

Die Auslastung oder der Auslastungsgrad [KRAR09] beschreibt die ausgenutzte Kapazität [KUTE93; GRO14]. Sie wird daher z.B. in [ELM91] auch *Kapazitätsausnutzung* genannt. Sie errechnet sich nach [ELM91] aus dem Verhältnis der Menge an produzierten Gütern in einem gewissen Zeitraum zur Kapazität innerhalb desselben Zeitraumes. Unter Berücksichtigung, dass mehrere Produkttypen innerhalb dieses Zeitraumes hergestellt werden können, kann folgende Metrik die auf jeden Produkttypen bezogene Auslastung beschreiben:

$$U_w = \frac{PM_w}{C_w \cdot t_p} \quad (3-14)$$

Mit:

$PM_w$ : Produzierte Menge des Werkstücktypen  $w$

$C_w$ : Produktionskapazität für Werkstücktyp  $w$

$t_p$ : Gesamtproduktionsdauer

Die Kennzahl (3-14) gibt das Verhältnis produzierter Produkte des Typen  $w$  zur maximalen Anzahl produzierbarer Produkte des Typen im selben Zeitraum an. Die Summe über alle Produkttypen kann als aggregierte Größe für die Gesamtauslastung verwendet werden. Im Falle paralleler Produktion steigt diese Kennzahl jedoch über 100% und das Maximum ist nicht bekannt, was die Interpretation erschwert. Als Alternative kann bspw. das Verhältnis zwischen minimaler Produktionsdauer zur mittleren Produktionsdauer eines Produkttypen herangezogen werden:

$$U'_w = \frac{t_w^{min}}{t_w^\mu} \quad (3-15)$$

Mit:

$t_w^{min}$ : Minimale Produktionsdauer für Werkstücke des Typen  $w$

$t_w^\mu$ : Mittlere Produktionsdauer für Werkstücke des Typen  $w$

Eine Auslastung kann auch für jede Ressource berechnet werden. Hierzu kann die Dauer des Ressourcen-Stillstandes zur Gesamtproduktionsdauer herangezogen werden.

### 3.3.1.14 Produktivität

Produktivität wird generell als das Verhältnis von Input zu Output verstanden, z.B. [AIK+08; BESÄ10; WEWA10]. Da jedoch kein einheitliches Verständnis dazu besteht, was als In- und Output verstanden wird, und speziell, wie diese zu messen sind [BESÄ10], gilt keine generelle Einigkeit zur Definition von Produktivität. Hier soll die Definition nach Jackson und Petersson [JAPÉ99] auf Basis von Zeitmessungen gelten. Für jede einzelne Maschine kann sie nach [JAPÉ99] wie folgt angegeben werden:

$$P_m = \frac{t_m^{HN}}{t_p} \quad (3-16)$$

Mit:

$t_m^{HN}$ : Hauptnutzungszeit der Maschine  $m$

$t_p$ : Gesamtproduktionsdauer

Für eine Ermittlung der Gesamtproduktivität einer Anlage sollten die einzelnen Produktivitäten addiert und durch die Summe der Maschinen dividiert werden. Somit ist gewährleistet, dass auch im Falle paralleler wertschöpfender Aktivitäten ein Wertebereich zwischen 0 und 1 erreicht werden kann.

### 3.3.2 Verfügbarkeit und Zuverlässigkeit

Verfügbarkeit und Zuverlässigkeit von Produktionssystemen sind bereits ausgiebig studiert (siehe bspw. [NAH93; BELE99; BIR99; MEPA02; AIK+08; SCH10; BRA15]). Im Folgenden findet eine kurze Darstellung der in dieser Arbeit verwendeten Metriken und zugrundeliegenden Zusammenhänge statt.

#### 3.3.2.1 Verfügbarkeit

Verfügbarkeit ist ein Merkmal zur Beschreibung der Wahrscheinlichkeit, dass ein System unter gegebenen Bedingungen an einem bestimmten Zeitpunkt seine Funktionalität ausführen kann [BIR99]. Die am weitesten verbreitete Metrik zur Bestimmung der Verfügbarkeit einer technischen Ressource ist die folgende (siehe z.B. [GES87; VDI 3649; BIR99; AIK+08]):

$$A_r = \frac{MTBF_r}{MTBF_r + MTTR_r} \quad (3-17)$$

Mit:

$MTBF_r$ : Mean Time Between Failures – Mittlere Dauer zwischen zwei Ausfällen einer technischen Ressource  $r$ <sup>21</sup>

$MTTR_r$ : Mean Time To Repair – Mittlere Dauer bis zur Betriebsbereitschaft nach einem Ausfall einer technischen Ressource  $r$

Die Verfügbarkeit gibt also den Anteil an ausfallfreier Zeit zur Gesamtproduktionszeit an. Je nachdem, wie ein Ausfall definiert ist, können hierbei verschiedene Verfügbarkeiten bestimmt werden, vgl. [AIK+08]. So ergibt sich bspw. eine niedrigere Verfügbarkeit, wenn nicht nur störungsbedingte Ausfälle und Instandsetzung berücksichtigt werden, sondern bspw. auch Ausfälle aufgrund von Inspektions- und Wartungsvorgängen [AIK+08].

Im Falle eines aus mehreren Komponenten (technischen Ressourcen) zusammengesetzten (Produktions-)Systems ist die Struktur wesentlich zur Bestimmung der Gesamtverfügbarkeit [Al 11]. Sind die Verfügbarkeiten einer Ressource bekannt, so ergibt sich die Verfügbarkeit von  $N$  seriell angeordneten Komponenten wie folgt:

$$A = \prod_{i=1}^N A_i \quad (3-18)$$

Sind die Verfügbarkeiten einer Ressource bekannt, so ergibt sich die Verfügbarkeit von  $N$  parallel angeordneten Komponenten wie folgt:

$$A = 1 - \prod_{i=1}^N (1 - A_i) \quad (3-19)$$

Zur vereinfachten Darstellung läuft in den Formeln (3-18) und (3-19) der Index  $i$  über alle der Serien- bzw. Parallelanordnung beteiligte Komponenten. Die angegebenen Beziehungen finden sich bspw. in [Ges87; AIK+08; Sch10]. Mischstrukturen können ebenfalls berechnet werden, z.B. indem Parallelstrukturen zusammengefasst werden und danach serielle Anordnungen zusammengerechnet werden. Weitere Möglichkeiten finden sich z.B. in [BIR99].

Meist wird nicht die Verfügbarkeit der technischen Ressourcen betrachtet, sondern die Verfügbarkeit ein Produkt in einem geforderten Durchsatz zu produzieren, vgl. [VDI 3649]. Hierbei wird auch von *Leistungsverfügbarkeit* gesprochen [VDI 4486]. Daher werden die Verfügbarkeit (und auch die Zuverlässigkeit) hier stets auf die Fähigkeit der Fertigung eines Produkttypen bezogen.

Zusammenfassend sind folgende Informationen notwendig für eine Verfügbarkeitsbetrachtung:

- Zeitpunkt und Dauer der Ausfälle der technischen Ressourcen (je nach vorhandener Information kann dabei ein Ausfall entsprechend definiert sein)
- Beteiligte technische Ressourcen zur Fertigung der verschiedenen Produkttypen sowie deren strukturelle Anordnung
- Mögliche alternative Materialflüsse zur Fertigung der Produkttypen

<sup>21</sup> In der Literatur häufig auch als MTTF – Mean Time To Failure bezeichnet

### 3.3.2.2 Zuverlässigkeit

Die Zuverlässigkeit eines technischen Systems beschreibt die Wahrscheinlichkeit, dass dieses unter gegebenen Bedingungen seine Funktionalität in einem gewissen Zeitintervall ausführen kann [BIR99]. Sie beschreibt die Fähigkeit in einem funktionstüchtigen Zustand zu bleiben. Im Gegensatz zur Verfügbarkeit bezieht sie sich nicht auf die Dauer von Ausfällen, sondern auf die Häufigkeiten dieser.

Die am weitesten verbreitete Metrik zur Zuverlässigkeit unterliegt der Annahme, dass die Ausfallrate des Systems oder einer Systemkomponente konstant ist, was für reale Systeme außerhalb der während des Großteils der Lebensdauer hinreichend valide ist [MEPA02]. Demnach lässt sich die Zuverlässigkeit wie folgt bestimmen (siehe z.B. [NAH93; BELE99; BIR99]):

$$Z_r(t) = e^{-\lambda_r t} \quad (3-20)$$

Mit:

$\lambda_r$ : Ausfallrate der technischen Ressource  $r$

Wesentlich zur Bestimmung der Ausfallrate sind die Ausfallzeitpunkte und deren jeweilige Dauer. Auch *MTBF* kann hieraus bestimmt werden. Eine entsprechende Vorgehensweise ist bspw. in [SCH10] aufgezeigt.

Analog zur Verfügbarkeit kann mithilfe der Kenntnis struktureller Zusammenhänge von Systemkomponenten eine Gesamtverfügbarkeit bestimmt werden [NAH93; BELE99; BIR99]. Die Formeln (3-18) und (3-19) gelten daher auch analog für die Zuverlässigkeit. Entsprechend sind zur Bestimmung der Zuverlässigkeit dieselben Informationen notwendig, wie für die Verfügbarkeit. Auch zur Bestimmung der Zuverlässigkeit sei empfohlen die Kennzahl auf die Fähigkeit zur Fertigung eines Produkttypen zu beziehen.

### 3.3.3 Flexibilität

Flexibilität wird allgemein als Fähigkeit eines Systems verstanden, sich auf ein wandelndes Umfeld anzupassen; unabhängig davon, ob diese Änderung vom System selber oder dessen Umgebung herbeigeführt wurde, vgl. [ZAMU07]. Insbesondere zur Handhabung von Änderungen ist Flexibilität eine wichtige Eigenschaft [VDF+14], die unter den zur Wandlungsfähigkeit beitragenden Merkmalen die stärkste Aussagekraft auf operativer Ebene hat, vgl. [SCH13]. Zudem ist Flexibilität eine wichtige Benchmark-Größe, die sich zur Bewertung sich ändernder Fertigungssysteme eignet [TAY97]. Sie ist des Weiteren ein relevantes Merkmal zur Bestimmung der Komplexität eines Fertigungssystems [REF90] mit Abhängigkeiten zur Performance einer Produktion [SEWI14].

Entsprechend findet sich eine Vielzahl an Literatur zur Definition und Operationalisierung von Flexibilität von Produktionssystemen, beginnend Anfang der 1980er Jahre bis heute, siehe bspw. [SDB+83; GER87; BASI88; BRMA89; GUGO89; REF90; SESE90; KONA92; SIWE92; KUTE93; CHCH96; TOT098; TSPH98; PAWI99; BMP+00; VOO'00; BEN01; ELM05; WAH05; CATE06; WWL08; ROG11; ZBF11; SEWI14]. Aufgrund ihres hohen Stellenwertes in der diskreten Fertigung (vgl. [TTV09; SEWI14]) und der Möglichkeit der Messbarkeit auf operativer Ebene (vgl. [ROG11])<sup>22</sup> wird die Flexibilität hier als zur Wandlungsfähigkeit beitragendes Merkmal vertieft behandelt.

Flexibilität wird üblicherweise in verschiedenen Teilflexibilitäten ausgedrückt. Die in der Literatur am häufigsten zitierte Quelle zur Definition dieser Teilflexibilitäten ist die Arbeit von Sethi und Sethi

<sup>22</sup> Aufgrund der vielseitig definierten Teilflexibilitäten kann Flexibilität im Allgemeinen auch zusätzlich der strategischen und der taktischen Ebene zugeordnet werden, vgl. [NYS+00].

[SESE90], die auch hier als Quelle für die informellen Definitionen der Teilflexibilitäten dient. Um Verwechslungen zu vermeiden, sind die Bezeichnungen der Teilflexibilitäten von Sethi und Sethi in Englisch übernommen. Deutsche Bezeichnungen weitestgehend gleich definierter Teilflexibilitäten sind z.B. in [REF90; KUTE93] zu finden. Aufgrund der hohen Anzahl an Arbeiten zu Metriken für diese Teilflexibilitäten und deren inhomogener Interpretation, vgl. [GMS+03], wird im Folgenden lediglich auf Teilflexibilitäten und Metriken eingegangen, deren Bestimmung auf Basis von Verhaltensbeobachtung möglich scheint und die in den diskutierten Betrachtungshorizont (Abschnitt 3.3) fallen (siehe auch [FÜL13<sup>6</sup>] und [LHF+13<sup>#</sup>]).

### 3.3.3.1 Machine Flexibility

Die *Machine Flexibility* bezieht sich auf die Anzahl an Operationen, die eine Maschine durchführen kann, ohne einen zu hohen Aufwand für den Wechsel der Operationen aufbringen zu müssen, vgl. [SESE90]. Machine Flexibility ist somit ein Maß für die Menge an Operationen einer Maschine. Eine aus dieser Definition direkt resultierende Metrik haben Chen und Chung gegeben [CHCH96]:

$$MF_m = \frac{|\Omega^m|}{|\Omega|} \quad (3-21)$$

Mit:

$|\Omega^m|$ : Anzahl der durch Maschine  $m$  durchführbaren Operationen

$|\Omega|$ : Anzahl aller durchführbaren Operationen des Fertigungssystems

Da nicht davon ausgegangen werden kann, dass die Fertigungsoperationen gleichverteilt genutzt werden und die Maschinen die Operationen mit derselben Effizienz ausführen, schlagen Chen und Chung des Weiteren eine von diesen Größen abhängige Kennzahl für die *Machine Flexibility* vor. Hier sei die Metrik von Brill und Mandelbaum [BRMA90] verwendet, die der von Chen und Chung entspricht, allerdings die Kennzahl auf einen Wertebereich [0,1] normiert:

$$MF'_m = \frac{\sum_{\omega=1}^{|\Omega|} g_\omega \cdot e_\omega^m}{\sum_{\omega=1}^{|\Omega|} g_\omega} \quad (3-22)$$

Mit:

$g_\omega$ : Gewichtung der Operation  $\omega$

$e_\omega^m$ : Effizienz, mit der Maschine  $m$  Operation  $\omega$  ausführen kann

Die Gewichtung einer Operation kann ermittelt werden als Verhältnis der Häufigkeit dieser Operation zur Gesamtanzahl durchgeführter Operationen. Die Effizienz der Operationen ist 0, wenn die Maschine die Operation nicht ausführen kann und entspricht dem Kehrwert der Dauer der Operation, wenn die Maschine sie durchführen kann (ggf. bei Bedarf normiert auf einen festen Wert).

Ähnlich zur Metrik (3-22), geben Wahab et al. [WWL08] eine auf einer sog. *efficiency Data Envelopment Analysis (DEA)* basierende Methode zur Bestimmung der *Machine Flexibility* an, die allerdings einer Mengenverteilung innerhalb eines Produktmixes und einer entsprechenden Verteilung der Operationen mehr Rechnung trägt. Hierzu wird zunächst für die zu untersuchende Maschine und für jeden Produkttypen und jede Operation ein Fähigkeitsindex bestimmt:

$$K_{mw\omega} = \begin{cases} 1, & \text{wenn für Produkttyp } w \text{ Operation } \omega \text{ an Maschine } m \text{ ausgeführt werden kann} \\ 0, & \text{sonst} \end{cases} \quad (3-23)$$

Aus den Fähigkeitsindizes wird dann eine *operative Fähigkeit* bestimmt, die ausdrückt, ob die Maschine bezogen auf einen Produkttypen einen Operations-Überschuss oder -Unterschuss besitzt:

$$\alpha_{mw} = \frac{\sum_{\omega \in \{\Omega^w \cap \Omega^m\}} K_{m\omega}}{|\Omega^w \cup \Omega^m|} \quad (3-24)$$

Mit:

$\Omega^w$ : Menge aller Operationen, die zur Herstellung von Produkten des Typen  $w$  benötigt werden

$\Omega^m$ : Menge der durch Maschine  $m$  durchführbaren Operationen

Hieraus kann nun die Wahrscheinlichkeit der Zuordnung von Produkttyp  $w$  zu Maschine  $m$  abgeschätzt werden:

$$p_{mw} = p_w \frac{\alpha_{mw}}{\max_{m \in M} \alpha_{mw}} \quad (3-25)$$

Mit:

$p_w$ : Wahrscheinlichkeit der Produktion von Produkttypen  $w$ , bzw. Anteil an Menge produzierter Produkte des Typen  $w$  ( $PM_w$ ) an Gesamtproduktionsmenge  $\sum_w PM_w$

Alternativ zu den Formeln (3-23), (3-24) und (3-25) kann  $p_{mw}$  auch aus Beobachtungen gewonnen werden. Hierzu wird ermittelt, wie häufig Werkstücke des Typen  $w$  der Maschine  $m$  zugeordnet wurden und dieser Wert wird in Verhältnis dazu gebracht, wie oft sie maximal hätten zugeordnet werden können.

Die *Machine Flexibility* ergibt sich dann wie folgt:

$$MF''_m = \sum_{w \in W} p_{mw} \sum_{\omega \in \{\Omega^w \cap \Omega^m\}} g_\omega \cdot e_{\omega w}^m \quad (3-26)$$

Mit:

$g_\omega$ : Gewichtung der Operation  $\omega$

$e_{\omega w}^m$ : Effizienz mit der Maschine  $m$  Operation  $\omega$  zur Bearbeitung von Werkstücktypen  $w$  ausführen kann

Die Gewichtung und Effizienz der Operationen können, wie für (3-22), aus der Häufigkeit bzw. der Dauer der Operationen bestimmt werden.

### 3.3.3.2 Material Handling Flexibility

Die *Material Handling Flexibility* gibt nach Sethi und Sethi Aufschluss über die Fähigkeit verschiedene Werkstücktypen durch das Fertigungssystem zu transportieren, vgl. [SESE90]. Diese Definition deckt sich mit der *Routing Flexibility* nach Nagarur [NAG92]<sup>23</sup>. Demnach wird gemäß folgender Metrik die Anzahl an Transportverbindungen zwischen den Maschinen im Verhältnis zur idealen maximalen Anzahl an Transportverbindungen zur Berechnung herangezogen:

<sup>23</sup> Sethi und Sethi haben eine andere Definition von Routing Flexibility, die im folgenden Abschnitt diskutiert wird

$$MHF = \frac{\sum_{i=1}^{|M|} \sum_{j=1}^{|M|} x_{ij}}{|M|(|M| - 1)} \quad (3-27)$$

Mit:

$|M|$ : Anzahl Maschinen im Fertigungssystem

$$x_{ij} = \begin{cases} 1, & \text{wenn Transportverbindung zwischen Maschine } i \text{ und Maschine } j \text{ existiert} \\ 0, & \text{sonst oder wenn } i = j \end{cases}$$

Die *Material Handling Flexibility* könnte auch pro Werkstücktyp relevant sein, falls die Werkstücktypen sich in Geometrie, Beschaffenheit und oder Gewicht stark unterscheiden und anzunehmen ist, dass nicht alle Werkstücktypen dieselben Transportwege nutzen können. In diesem Fall ist  $x_{ij}$  jeweils auf den zu untersuchenden Werkstücktypen zu beziehen.

### 3.3.3.3 Routing Flexibility

Die Routing Flexibility gibt an, wie viele Routen ein Werkstück zur Produktion durch das Fertigungssystem nehmen kann [SESE90]. Eine direkt aus dieser Definition folgende Metrik bezogen auf einen Werkstücktypen haben Bernardo und Mohamed [BEMO92] veröffentlicht:

$$RF_w = 1 - \frac{1}{AR_w} \quad (3-28)$$

Mit:

$AR_w$ : Anzahl der Routen, die Werkstücktyp  $w$  durch das Fertigungssystem nehmen kann

### 3.3.3.4 Operation Flexibility

Die *Operation Flexibility* ergibt sich aus der Anzahl unterschiedlicher Prozessfolgen zur Produktion eines Produktes [SESE90]. Sie berücksichtigt entsprechend die Menge an unterschiedlichen Operationsreihenfolgen und die Möglichkeit Operationen durch andere zu substituieren, um einen bestimmten Produkttypen zu fertigen. Ähnlich wie bei der *Material Handling Flexibility* kann zur Berechnung ein Optimum herangezogen werden, welches bei einer bestimmten Anzahl an Operationen für einen Werkstücktypen alle möglichen Mengen und Reihenfolgen an Operationen erlaubt [FÜL13<sup>9</sup>]:

$$OF_w = \frac{N_w}{|\Omega^w|! \cdot 2^{|\Omega^w|}} \quad (3-29)$$

Mit:

$N_w$ : Anzahl an Prozessfolgen (Arbeitsplänen) zur Fertigung von Produkttyp  $w$

$|\Omega^w|$ : Anzahl der zur Fertigung von Produkttyp  $w$  erforderlichen Operationen

Das zur Normierung herangezogene Optimum im Nenner von (3-29) sieht zum einen vor, dass jede Reihenfolge an Operationen möglich ist ( $|\Omega^w|!$ ). Zum anderen wird angenommen, dass jede Operation von einer beliebigen anderen Operation zur Herstellung des Produkttypen substituiert werden kann ( $2^{|\Omega^w|}$ ).

### 3.3.3.5 Process Flexibility

Nach Sethi und Sethi beschreibt die *Process Flexibility* die Menge an Produkttypen, die das Fertigungssystem ohne größeren Umrüstaufwand produzieren kann, vgl. [SESE90]. Sie ist also ein Maß für die

Bandbreite an produzierbaren Produkten. Diese Definition deckt sich mit der *Product Mix Flexibility Response* nach Wahab [WAH05], der diese als Fähigkeit beschreibt zwischen der Produktion verschiedener Produkte zu wechseln. Daher soll die von Wahab definierte Metrik zur Ermittlung der *Product Mix Flexibility* hier für die *Process Flexibility* gelten:

$$PF = \frac{1}{|M|} \sum_{m=1}^{|M|} \sum_{w=1}^{|W|} \sum_{i=m}^{|M|} p_{mw} a_{wi} \quad (3-30)$$

Mit:

$|M|$ : Anzahl Maschinen im Fertigungssystem

$|W|$ : Anzahl unterschiedlicher Produkttypen

$p_{mw}$ : Wahrscheinlichkeit der Zuordnung eines Werkstückes des Produkttypen  $w$  zur Bearbeitung an Maschine  $m$  (zu ermitteln nach Formel (3-25) oder dem tatsächlichen Anteil an Zuordnungen im Verhältnis zur Gesamtanzahl produzierter Produkte des Typen  $w$ )

Der Parameter  $a_{wi}$  entspricht einem Maß dafür, wie viele Operationen<sup>24</sup> sowohl für Produkttyp  $w$  als auch für Produkttyp  $i$  genutzt werden können:

$$a_{wi} = \frac{|\Omega^w \cap \Omega^i|}{|\Omega^w \cup \Omega^i|} \quad (3-31)$$

Mit:

$\Omega^w$ : Menge aller Operationen, die zur Herstellung von Produkttyp  $w$  (bzw.  $i$ ) benötigt werden

---

<sup>24</sup> Wahab verwendet ursprünglich die Menge an Werkzeugen, die notwendig sind. Eine Betrachtung der notwendigen Operationen wird hier jedoch als sinnvoller erachtet, da sie der Kennzahl einen allgemeineren Charakter verleiht und die Möglichkeit zur Ermittlung aus Prozessdaten wahrscheinlicher scheint.

## 4 Beschreibungsmittel zur Verhaltensmodellierung von Fertigungsanlagen

Ein Beschreibungsmittel dient der (formalen) Formulierung (Modellierung) von Aufgaben und Lösungen eines Problems [SCH99]. Ein entstandenes Modell entspricht dabei stets einer Abstraktion der realen Welt und soll zielgerichtet, verständlich, eindeutig und widerspruchsfrei bestimmte Sachverhalte und Zusammenhänge abbilden [ASS97].

Nach Schnieder [SCH99] wird das *Verhalten* eines Systems anhand seines *Zustandsverlaufs* beschrieben. Ein *Zustand* bzw. die abstrakte Zustandsgröße wird durch ein Attribut-Wertemengen-Paar beschrieben. Ein Attribut charakterisiert den Typ der Zustandsgröße, der Wert einer Zustandsgröße ist dessen *Elementarzustand*. Aus der Menge der Elementarzustände aller Zustandsgrößen setzt sich der *System- oder Globalzustand* zusammen. Der Wechsel eines Zustands ist als *Ereignis (Event)* definiert [SCH99]. Dieses wird i.d.R. als zeitlos angenommen, d.h. der Zustandswechsel hat keine Dauer [LUN09]. Das *Verhalten* des Systems ergibt sich aus der Folge von Zuständen des Systems über der Zeit. Sprachentheoretisch betrachtet (z.B. [SEA13]), wird ein Elementarzustand *Zeichen* genannt. Die Menge aller möglichen Zeichen entsprechen dem *Alphabet* des Systems und der Globalzustand wird als *Wort* bezeichnet. Bei der Menge aller im Verhalten (möglich) aufkommenden Wörter handelt es sich um die *Sprache* des Systems.

Im Folgenden werden gängige Beschreibungsmittel diskutiert, die sich zur Modellierung des Verhaltens von diskreten Fertigungsanlagen eignen. Darüber hinaus eignen sie sich zur Gewerke-übergreifenden Beschreibung des Systemverhaltens (d.h. des interdisziplinären immanenten Verhaltens). Die diskutierten Beschreibungssprachen sind Methoden- und Werkzeug-neutral gewählt sowie in der Literatur weit verbreitet und/oder in Normen o.ä. standardisiert. Dies erhöht die Wahrscheinlichkeit, dass ein für den Ansatz zu wählendes Beschreibungsmittel allgemein bekannt ist und zur generellen Dokumentation sich ändernden Verhaltens herangezogen werden kann.

### 4.1.1 Zustandsautomaten

Verhaltensmodellierung kann u.a. mit sog. Kripke-Strukturen, d.h. Graphen, in denen die Knoten Systemzuständen und die Kanten Zustandsübergängen entsprechen, vorgenommen werden [DIOL03]. Hierzu zählen die Zustandsautomaten. Ein initialisierter Zustandsautomat  $\mathcal{A}$  kann nach [LUN09] wie folgt beschrieben werden:

$$\mathcal{A} = (\mathcal{Z}, \mathcal{G}, z_0) \quad (4-1)$$

Mit:

$\mathcal{Z}$ : Menge der Systemzustände bzw. Sprache des Systems

$\mathcal{G}$ : Menge der Zustandsübergangsfunktionen

$z_0$ : Initialzustand

Die Menge aller möglichen Zustandsübergänge, resp. das Verhalten des Systems, sind durch die Zustandsübergangsfunktionen beschrieben. Diese können bspw. in tabellarischer Form oder i.F.v. booleschen Ausdrücken oder Logikfunktionen vorliegen, vgl. [LUN09]. Sie beschreiben die Kanten des Automatengraphen, während die Zustände den Knoten des Automatengraphen entsprechen.

Zustandsautomaten bieten, aufgrund der Beschreibung als Graphen, die Möglichkeit einer leicht verständlichen graphischen Darstellung. Sie sind ebenso befähigt deterministisches wie stochastisches und/oder zeitbehaftetes Verhalten darzustellen, vgl. [LUN08; BÉR13]. Dazu sind sie mathematisch-formal und können z.B. in Form einer Zustandsraumdarstellung oder mittels Zustandsübergangsmatrix darge-

stellt werden und sind entsprechend maschinenlesbar. Es stehen viele Analysealgorithmen u.a. aus der Graphentheorie zur Verfügung, die es erlauben, formale Eigenschaften (z.B. Erreichbarkeit, Reversibilität, Lebendigkeit) der Modelle zu analysieren, vgl. [DIÉ00]. Ein Nachteil der Automaten ist jedoch dem geschuldet, dass technische Systeme häufig hoch nebenläufiges Verhalten aufweisen. Dies führt typischerweise zu einem exponentiellen Anstieg an Zuständen (*Zustandsexplosion*) [VDI/VDE 3681].

#### 4.1.2 Petri-Netze

Petri-Netze [PET62] entsprechen, ebenfalls wie Automaten, Kripke-Strukturen [DIOL03]. Allerdings handelt es sich hierbei um bipartite Graphen. Die Knoten können Stellen  $\mathcal{S}$  oder Transitionen  $\mathcal{T}$  sein und sind mittels gerichteter Kanten verbunden, die durch Flussrelationen  $\mathcal{F} \subseteq (\mathcal{S} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S})$  beschrieben sind.

Um den aktuellen Zustand eines Petri-Netzes darzustellen, können Stellen mit *Marken* versehen sein. Ein Vektor, der für jede Stelle beschreibt, ob sie mit einer (oder mehrerer) Marke(n) versehen ist (und ggf. mit welcher „Art“ Marke), wird *Markierung*  $m$  genannt. Die Marken können durch das Petri-Netz wandern, was einer Zustandsänderung entspricht. Formal ist die Beschreibung des Petri-Netzes  $\mathcal{PN}$ :

$$\mathcal{PN} = \langle \mathcal{S}, \mathcal{T}, \mathcal{F}, m_0 \rangle \quad (4-2)$$

Mit:

$\mathcal{S}$ : Menge der Stellen

$\mathcal{T}$ : Menge der Transitionen

$\mathcal{F}$ : Menge der Flussrelationen

$m_0$ : Initialmarkierung

Die einfachste Form von Petri-Netzen mit Markierung sind die sog. *Bedingungs-Ereignis-Netze* (B/E-Netz) [ABE90]. Sie sind entsprechend Formel (4-2) definiert und dadurch charakterisiert, dass die Markierung einer Stelle binär ist. Auf den B/E-Netzen aufbauend, existiert eine Vielzahl weiterer allgemeiner Petri-Netz Typen. Beispiele sind *stochastische* und *zeitbehaftete* Petri-Netze (vgl. [DAAL10]), *Stellen-Transitions-Netze* [ABE90], *gefärbte Petri-Netze* [DAAL10], *kontinuierliche* oder *hybride Petri-Netze* [CDD+00; DAAL10] sowie *interpretierbare Petri-Netze*, die sich bspw. zum Steuerungsentwurf eignen (*Steuerungsinterpretierbare Petri-Netze*) [KFL02]. Zudem existiert eine Vielzahl spezialisierter Petri-Netz-Typen für bestimmte Anwendungsfälle, vgl. [MOGU96].

Neben einer grafischen Darstellung können Petri-Netze mathematisch in Form einer *Inzidenzmatrix*  $\mathcal{C}$  beschrieben werden. Zur Analyse von Petri-Netzen kann ein Erreichbarkeitsgraph aus dem Petri-Netz und seiner Initialmarkierung generiert werden. Er beschreibt alle möglichen Zustandsverläufe eines Petri-Netzes und entspricht einem Zustandsautomaten mit selbigem Verhalten. Daher lassen sich Zustandsautomaten auch in Petri-Netzen darstellen [MUR89] und Petri-Netze in Zustandsautomaten transformieren [CGS06]. Es existiert eine Vielzahl an formalen Analysen für Petri-Netze zur Ermittlung von Eigenschaften wie Deadlock-Freiheit, erreichbaren Zuständen oder Lebendigkeit, vgl. [MUR89]. Petri-Netze eignen sich besonders zur Darstellung paralleler Aktivitäten bzw. Nebenläufigkeiten [MUR89]. Aufgrund ihrer Universalität finden Petri-Netze Anwendung in unterschiedlichsten Domänen, vgl. [CDD+00]. Auch in der Fertigungsautomation [SCH99] und der Analyse von Fertigungssystemen [DHP+93; MOGU96] finden sie häufig Anwendung. Ihre graphische Darstellung in Form des Netzes sowie in Form des Erreichbarkeitsgraphen erlaubt eine einfache Interpretierbarkeit durch den Menschen. Daher haben sie einen starken Dokumentationscharakter.

### 4.1.3 UML und SysML

Aus dem Bereich der objektorientierten Softwareentwicklung und dem Bedarf einer hierfür einheitlichen, universellen und leicht verständlichen Modellierungssprache entstand die *Unified Modelling Language (UML)* [Oes09], die in der internationalen Normenreihe [ISO/IEC 19505] standardisiert ist. Eine auf das Systems Engineering spezialisierte Erweiterung der UML nennt sich *Systems Modeling Language (SysML)* [Obj12]. SysML setzt sich aus acht verschiedenen Diagrammtypen zusammen. Hiervon sind die folgenden vier als Verhaltensdiagramme klassifiziert:

1. Aktivitätsdiagramm: Darstellung von Aktivitäten mithilfe von Objekt- und Kontrollflüssen
2. Sequenzdiagramm: Darstellung der sequenziellen Interaktion zwischen Objekten
3. Zustandsdiagramm: Zustandsdarstellung ähnlich des Zustandsautomaten
4. Anwendungsfalldiagramm (Use-Case-Diagramm): Modellierung von Anwendungsfällen zur Spezifikation von (Software-Systemen)

Das Sequenzdiagramm ist ein sog. *Interaktionsdiagramm* [ISO/IEC 19505], d.h. es bildet die Interaktion zwischen Objekten ab. Relevant in dieser Arbeit ist jedoch das interne zustandsbasierte Verhalten, weshalb das Sequenzdiagramm nicht weiter berücksichtigt wird. Das Anwendungsfalldiagramm entspricht einer reinen Spezifikationssprache und wird auf einem sehr hohen Abstraktionsniveau modelliert. Es ist daher stark informell und soll im Folgenden auch nicht weiter berücksichtigt werden. Aktivitäts- und Zustandsdiagramm werden in den folgenden Abschnitten (4.1.3.1 und 4.1.3.2) erläutert.

#### 4.1.3.1 Aktivitätsdiagramm

Eine *Aktivität* spezifiziert in UML und SysML ein ihr untergeordnetes Verhalten mittels Daten- und Kontrollflüssen [ISO/IEC 19505]. Zur Datenmanipulation können Aktivitäten unterschiedliche Arten von *Aktionen* beinhalten [ISO/IEC 19505]. Aktionen werden in Form von Knoten dargestellt, die Daten- und Kontrollflüsse zwischen den Aktionen werden als Kanten dargestellt. Über die Kanten können, ähnlich wie bei Petri-Netzen, *Marken* zwischen den Aktionen fließen. Eine Marke ist in jedem Diagramm ein eindeutiges Element und kann ein Objekt, ein Datum oder ein Steuersignal repräsentieren. Marken befinden sich stets an Aktionen und können von ihnen verbraucht und/oder generiert werden. Zudem können Aktivitäten mit Parametern versehen werden. Um die Markenflüsse durch eine Aktivität zu lenken, stehen verschiedene Knoten wie Verzweigungen, Synchronisationsknoten, Entscheidungs- und Vereinigungsknoten zur Verfügung.

Das Markenkonzept und die flussstrukturierenden Elemente erlauben eine effiziente Modellierung von Nebenläufigkeiten, ähnlich wie mit Petri-Netzen. Die Möglichkeit Signale zwischen Aktivitäten zu senden, erlaubt außerdem eine modulare Modellierung. Zudem kann eine Modellierung mit hohem Abstraktionsgrad vorgenommen werden. All dies erhöht die Lesbarkeit und Interpretierbarkeit durch den Menschen. Die Elemente besitzen zwar eine eindeutige Semantik, allerdings fehlen UML und SysML generell eine mathematisch-formale Basis [KKP+99]. Daher werden UML und SysML *als semi-formale Beschreibungsmittel* klassifiziert [VDI/VDE 3681]. Der Möglichkeit eines hohen Abstraktionsgrades und damit einer leichten Verständlichkeit steht somit ein fehlender Formalismus gegenüber.

#### 4.1.3.2 Zustandsdiagramm

Das Zustandsdiagramm beinhaltet sog. *Zustandsmaschinen (State Charts)*, die sehr stark an Automaten angelehnt sind. Die UML hat damit eine Beschreibungssprache aufgenommen, die auch außerhalb des Kontextes von UML u.a. in der Automatisierungstechnik genutzt wurde, vgl. [Sch99].

Eine Zustandsmaschine besteht aus *Zuständen* und *Zustandsübergängen*. Wie bei Automaten werden diese mittels Knoten und Kanten dargestellt und es kann stets nur ein Zustand innerhalb einer Zustandsmaschine aktiv sein. Zustandsübergänge sind mit *Ereignissen* annotiert, die den Zustandswechsel hervorrufen. Das Ereignis kann dabei beliebig abstrakt und mit Parametern versehen sein. Dazu können *Aktionen* an den Zustandsübergängen annotiert werden, die beim entsprechenden Zustandswechsel ausgeführt werden. Über Aktionen ist es möglich, verschiedene Zustandsautomaten miteinander „kommunizieren“ zu lassen, was die Modellierung von Nebenläufigkeiten und verteilter Systeme erlaubt [OES09]. Zustandsmaschinen können hierarchisch modelliert werden, was die Modellierung verschiedener Detaillierungs- und Abstraktionsgrade erlaubt. Dies erhöht die Lesbarkeit und Verständlichkeit und daher den Dokumentationscharakter. Jedoch gilt auch für die Zustandsmaschinen, dass sie ein semi-formales Beschreibungsmittel sind. UML schreibt nicht vor, das Verhalten von Aktionen und dessen Auswirkungen genauer zu spezifizieren. Dies kann zu unbekanntem Verhalten führen.

#### 4.1.4 Algebraische Modelle

Algebraische Modelle können zur Darstellung von Fertigungsprozessen mithilfe mathematischer Gleichungen verwendet werden. Hierzu werden meist simple Modelle (z.B. einzelner Arbeitsstationen) mittels ihrer physikalischen Eigenschaften und Abhängigkeiten hergeleitet und zu komplexen Modellen miteinander verknüpft [CuFe10]. Die Modelle unterliegen meist verschiedenen generellen und speziellen Annahmen und es existieren entsprechend viele Modelle zur Beschreibung gleicher Systeme, vgl. [DAG92]. Auch boolesche Algebra oder temporale Logik eignen sich zur Modellierung von diskreten Zusammenhängen [VDI 2221]. Ein Beispiel für ein algebraisches Modell ist die (diskrete) Zustandsraumdarstellung [LUN08].

Zur Modellierung von Fertigungssystemen mittels algebraischer Modelle ist ein tiefes Verständnis der zugrundeliegenden physikalischen und logischen Prinzipien notwendig. Zudem setzen Modellierung und Analyse ein tiefes mathematisches Verständnis voraus. Algebraische Modelle werden schnell sehr groß (bzgl. der Menge an Variablen und Gleichungen) und komplex. Sie sind somit für den Menschen schwer verständlich und damit weniger für eine Dokumentation des Verhaltens geeignet als grafische Modelle. Ihr Vorteil liegt allerdings im strikten Formalismus sowie den genauen Analyseergebnissen [CuFe10].

#### 4.1.5 Process Specification Language

Die Process Specification Language (PSL) [NAT02<sup>@</sup>] dient der Beschreibung einer einheitlichen Semantik mittels einer Ontologie und darauf basierenden Modellen mit dem Ziel des Informationsaustausches zwischen verschiedenen Disziplinen der Fertigung, wie z.B. der Prozessmodellierung, Produktionsplanung oder Simulation [GRME03]. Als Syntax verwendet die PSL Prädikatenlogik. Mithilfe dieser können Axiome zur Beschreibung von Relationen und möglichen Funktionen zwischen definierten Begriffen beschrieben werden [CHR15]. Die PSL ist in der [ISO 18629] standardisiert und bietet u.a. die Möglichkeit, die diskreten Zustände und deren Übergänge und somit das Verhalten eines Fertigungssystems zu beschreiben. Die Beschreibung des Verhaltens eines Systems kann jedoch schnell sehr komplex werden. Der Generalität der Beschreibungssprache sowie der Möglichkeit der Schlussfolgerung mittels Inferenz (wie dies bei allen Ontologien der Fall ist) steht ein hoher Modellierungsaufwand gegenüber. Zudem ist die Darstellungsweise aufgrund der Nutzung der Prädikatenlogik als Syntax nur schwer verständlich.

## 5 Bewertung evolvierender Fertigungssysteme – Stand der Forschung und industrielle Praxis

Anlagenevolution fand lange Zeit keine Beachtung in der Forschung, da der Fokus auf der Anlagenplanung als „Greenfield“-Projekt lag [StFA11]. Eine Vielzahl existierender Ansätze beschäftigt sich daher mit Produktions- und Automationsparadigmen, die Anlagen wandlungsfähig machen, vgl. [MUK+02; ELM05; WEN+07; ALE11]. Dazu zählen die *Flexiblen Fertigungssysteme* [SBK06], *Rekonfigurierbaren Fertigungssysteme* [MUK00; DAS03], *Evolvierbaren Produktionssysteme* [RBC+10; CCB+11] sowie *Bionische und Holonische Produktionssysteme*, vgl. [TWN98]. Auch Kombinationen verschiedener Paradigmen wurden untersucht, vgl. [TTV09]. Allerdings sind diese Ansätze nicht auf Bestandsanlagen anwendbar und betrachten selten die Bewertung von Änderungen. Im Folgenden sollen Arten von Ansätzen vorgestellt werden, die bei der Evolution von bestehenden Fertigungssystemen unterstützend eingesetzt werden können und eine Bewertung von Änderungen beinhalten oder begünstigen. Im Speziellen sind dies modellbasierte Ansätze (Abschnitt 5.1) sowie modellernende Ansätze (Abschnitt 5.2). Der Fokus steht hierbei immer auf der Dokumentation und Evaluierung vorgenommener oder vorzunehmender Änderungen. Zudem werden die in der Industrie vorgefundene Praxis im Umgang mit Anlagenevolution und deren Gründe diskutiert (Abschnitt 5.2.3).

### 5.1 Modellbasierte Ansätze

Die Nutzung von Modellen zum Entwurf von Software oder technischen Systemen wird auch als *modellbasiertes Engineering* (*Model Based Engineering, MBE*) bezeichnet. Die [VDI 2206] beschreibt den modellbasierten Entwurf von mechatronischen Systemen als Phasenmodell (Abbildung 5-1).

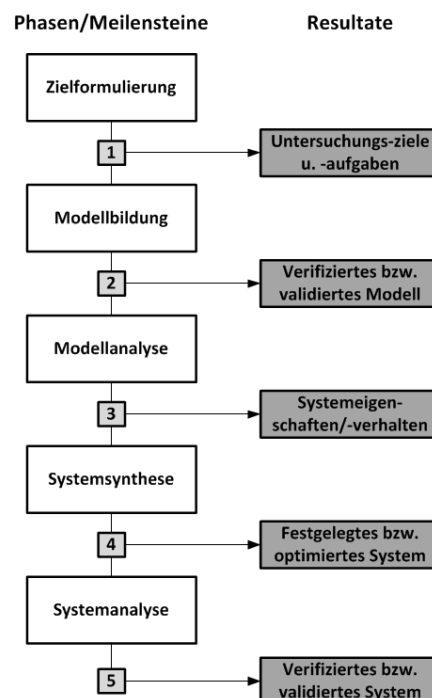


Abbildung 5-1: Modellgetriebener Systementwurf nach [VDI 2206]

Grundlage des Phasenmodells ist stets die Beschreibung der an das System gestellten Anforderungen (Zielformulierung). Darauf aufbauend findet eine Modellbildung des Systems statt. Benötigte Parameter und Zusammenhänge können hierfür geschätzt, geplant oder experimentell erhoben (im Falle eines

bereits existierenden Systems) sein. Das erstellte Modell kann dann verifiziert und validiert werden. Dies geschieht im optimalen Fall automatisiert, sofern das Modell einen entsprechenden Formalisierungsgrad aufweist. Ist die Modellbewertung positiv ausgefallen, so wird das Modell als Spezifikation zur Systemsynthese genutzt. Das Vorgehen lehnt sich somit an das V-Modell an (Abschnitt 2.2). Vorteile sind die Bewertung einer Systemlösung, bevor sie realisiert wird, sowie die Möglichkeit die erstellten Modelle zur Dokumentation des Systems zu nutzen und zur Evolution des Systems weiter nutzen zu können.

Viele aktuelle MBE-Ansätze verwenden Spezifikationssprachen wie UML [BLD00; MCC05; T.G.06], SysML [BSB+11; VSF+14], Derivate bzw. Dialekte von UML oder SysML (z.B. *UML4Mechatronics*, *plcML*, vgl. [DMB+10; VFS+15]) oder proprietäre Modellierungssprachen, die eine multi-disziplinäre Modellierung erlauben, z.B. [WML99]. Um einen zur automatischen Analyse notwendigen Formalisierungsgrad zu erlangen<sup>25</sup>, verfeinern oder formalisieren die Ansätze die UML- bzw. SysML-Semantik [VYA13]. Andere Ansätze transformieren UML/SysML in formale Modelle, z.B. [BLD00; MCC05]. Ein Hauptaugenmerk neuerer multi-disziplinärer Ansätze liegt häufig auf der Konsistenzprüfung zwischen den Disziplinen [LMC+14; VFS+15]. Im Falle von Systemevolution hilft dies insbesondere in der Erkennung der Propagation einer Änderung von einer Disziplin zur anderen [HCL+14].

Eine Herausforderung modellbasierter Ansätze im Umfeld sich ändernder Produktionssysteme liegt in der Tatsache, dass die Modelle sich genauso stetig ändern müssen, wie die Systeme selbst [BSK+14]. Dies wird als *Co-Evolution* zwischen Modell und realem System bezeichnet [VFF+15<sup>#</sup>]. Ansätze verfolgen daher die Kombination aus Änderungsmanagement und MBE [LSP+15] oder Versions- und Variantenmanagement und MBE [BBO+12] zur Evolutionsunterstützung. Einige Ansätze beschäftigen sich mit der Nutzung von Ontologien zur modellgestützten Anlagenmodernisierung [SFT+10; STFA11; SRF+11], Co-Evolution zwischen unterschiedlichen Engineering Disziplinen [LSF+14] oder zur Analyse von Änderungseinflüssen auf verwendete Modelle [FKV14]. Diese Ansätze gehen stets von einem bestehenden Modell aus. Daher beschäftigen einige Vorhaben sich auch mit der Ableitung einer Systemlösung aus den Anforderungen sowie dem Vorgehen zur Detaillierung der Modelle, z.B. [T.G.06; BSB+11]. Allgemein kann ausgesagt werden, dass sich modellbasierte Ansätze zur Unterstützung von Anlagenevolution und insbesondere auch zur Bewertung vorzunehmender und vorgenommener Änderungen eignen. Daher werden im Folgenden modellbasierte Methoden zur Analyse von Fertigungssystemen und deren Anwendbarkeit zur Bewertung von Evolution diskutiert. Diese können im Rahmen eines MBE-Ansatzes Anwendung finden, aber auch für sich stehen und z.B. der nachträglichen Analyse eines Bestandssystems dienen.

### 5.1.1 Modellbasierte Analyse und Bewertung von Fertigungssystemen

Ansätze zur Analyse von Fertigungssystemen mittels algebraischer Modelle existieren in hoher Anzahl, vgl. [DAGE92]. Entsprechend viele Modellierungen existieren u.a. zur Untersuchung der Performance (z.B. [NWW97; CuFe10]), Flexibilität (z.B. [CHTO92; NAG92]) oder zur Fehleranalyse (z.B. [FTK16]). Typisch ist hierbei die Analyse der Auswirkung von Auftragsschwankungen, Maschinenausfällen, Werkzeugwechselzeiten oder Wartungsarbeiten auf entsprechende Indikatoren des Fertigungssystems [DAGE92].

Die genutzten Modelle können sehr präzise Analyseergebnisse und exakte Größen liefern, vgl. [DAGE92; CuFe10]. Dem gegenüber steht allerdings, dass die Modelle meist viele Eingangsparameter und einen geringen Generalitätscharakter aufgrund einer Vielzahl von Annahmen besitzen [DAGE92]. Des Weiteren sind die algebraischen Modelle wenig abstrakt und daher schwer zu handhaben sowie schwer interpretierbar. Entsprechend aufwändig ist auch die Modellierung. Im Falle statistisch verteilter Größen wird eine Parametermittlung durchgeführt. Die Ergebnisse können in ungünstig auftretenden Parameter-

<sup>25</sup> Häufig verwenden MBE-Ansätze auch Code-Generierung, wofür mathematisch-formale Modelle notwendig sind.

Konstellationen jedoch stark von den reellen Werten abweichen [ZIWE06]. Daher werden algebraische Modelle häufig zur Simulation eingesetzt, vgl. [ZIWE06]. Simulation erlaubt die Wiederverwendung und neu-Parametrierung einzelner Modellfragmente oder des gesamten Modells und somit die Analyse des Systems unter unbekanntem Parametern oder Einsatzszenarien [VISR11]. Der Nutzen einer Simulation hängt allerdings von der Wahl des Simulationsszenarios ab [VDI 2221]. Zudem leidet Simulation darunter, dass seltene Ereignisse auch in der Simulation selten berücksichtigt werden; diese können jedoch eine hohe Auswirkung auf das System und seine Performance haben [VFS+15].

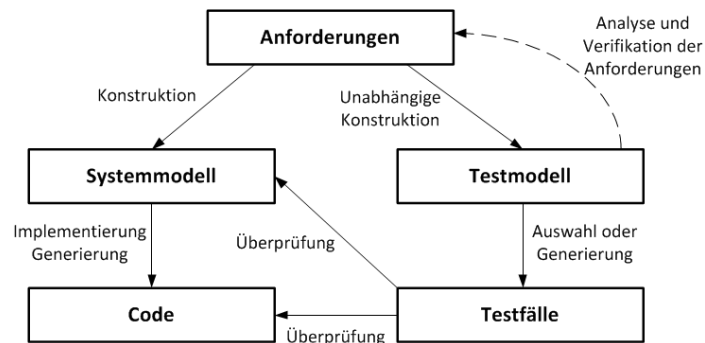
Im Gegensatz zu algebraischen Modellen zeichnen sich grafische Modelle durch eine einfache und leicht verständliche Darstellungsweise der Zusammenhänge eines Systems aus [CHR15]. Aufgrund dessen eignen sie sich eher zum Entwurf von Fertigungssystemen [DHP+93] und dienen als gutes Dokumentationsmittel. Formale grafische Modelle eignen sich genauso wie algebraische Modelle zur Simulation. Zur Analyse von formalen Eigenschaften von Fertigungssystemen werden meist Methoden der Graphentheorie eingesetzt [DHP+93]. Ein häufig verwendetes Mittel zur Überprüfung vorgegebener Eigenschaften aus formalen Modellen ist das *Model-Checking*, vgl. [CGP01; KLE09]. Hierzu müssen die zu überprüfenden Eigenschaften in Form von Formeln beschrieben sein [CAMA09]. Meist werden diese in temporaler Logik beschrieben, vgl. [DIOL03; CAMA09]. Bestimmte Ereignisse können somit auf ihr Erscheinen, ihre Nicht-Existenz, Abhängigkeit untereinander oder eine bestimmte Abfolge bzw. Kausalität hin überprüft werden, vgl. [CAMA09]. Dies eignet sich zur Überprüfung allgemeiner Aussagen über das Systemverhalten oder von spezifisch formulierten Anforderungen [DIOL03]. So können z.B. Sicherheitsanforderungen [PEFA13] oder funktionale Anforderungen formuliert werden [KFL02; HCL+14]. Das Modell kann, nach Formulierung der Anforderungen, dahingehend überprüft werden, ob es die spezifizierten Eigenschaften enthält. Model-Checking hat den Vorteil, dass alle möglichen Systemzustände berücksichtigt werden [VFS+15]. Die Formulierung der Anforderung ist jedoch ein nicht-trivialer Schritt und fehleranfällig. Dieser Schritt erfordert Verständnis für das Modell, die zu prüfende Anforderung und die Logik, in der die Eigenschaft ausgedrückt werden soll [CAMA09]. Der Transfer von der Formel in temporaler Logik hin zu der eigentlichen abstrakten Eigenschaft ist dem Nutzer überlassen. Er muss also alle Anforderungen kennen und diese auch entsprechend korrekt ausdrücken können. Dieser Vorgang ist daher sehr fehleranfällig [KLE09]. Zudem eignen sich die Ausdrücke selten zur Überprüfung nicht-funktionaler Eigenschaften [VFS+15]. Auch qualitative Eigenschaften wie Performance und Flexibilität werden mithilfe von graphenbasierten formalen Modellen ermittelt. Beispiele sind Performance Analysen [DHP+93], die Bestimmung von Flexibilitäten [BASi88; BASi90] oder die Verifikation von Sicherheitsanforderungen [SPJ+03]. Hierfür ist jedoch eine speziell dafür ausgerichtete Modellierung notwendig.

Eine weitere Bewertungs-Methode ist das *modellbasierte Testen (MBT)* [LIG09]. Testen dient der Überprüfung der Übereinstimmung von Anforderungen und Verhalten einer Software oder eines Systems i.S.v. gleichem Eingangs-/Ausgangsverhalten [RUP+15]. Hierzu werden sog. *Testfälle* (Test Cases) erstellt, die dieses in einer *Testdurchführung* überprüfen. Ein Testfall entspricht einer Menge an Testeingaben, Ausführungsbedingungen und erwarteten Systemausgaben zur Überprüfung von Anforderungen [XIO11]. Die manuelle Testfall-Erstellung aus einer informellen Spezifikation ist allerdings sehr aufwändig und führt häufig zu einer unzureichenden Testabdeckung [MKD15]. MBT erleichtert die Erstellung von Testfällen [LIG09]. Das prinzipielle Vorgehen ist in Abbildung 5-2 dargestellt. Aus den gegebenen Anforderungen werden ein Systemmodell und, unabhängig davon<sup>26</sup>, ein Testmodell erstellt. Das Testmodell entspricht einer modellbasierten Darstellung der zu erstellenden Testfälle. Automaten eignen sich hier-

---

<sup>26</sup> Es besteht auch die Möglichkeit ein kombiniertes Test- und Systemmodell zu verwenden, die dargestellte Vorgehensweise entspricht jedoch aufgrund der Unabhängigkeit von Testfallgenerierung und Code-Implementierung der empfohlenen, vgl. [LIG09].

für [LIG09] ebenso wie Petri-Netze [KRA12]. UML-Zustandsdiagramme sind ebenfalls ein beliebtes Beschreibungsmittel zur Ableitung von Testfällen [LIG09; KRA12], aber auch andere UML-Diagramme, z.B. Sequenzdiagramme [HWO+10; VKT+13; MKD15] oder Zeitdiagramme [RTS+14; RFT15] können genutzt werden. Aus dem Testmodell werden Testfälle generiert oder ausgewählt. Aus dem Systemmodell wird Code generiert oder es wird als Spezifikation für eine manuelle Implementierung genutzt. Das Vorgehen entspricht folglich dem des MBE. Im Anschluss können die Testfälle auf dem Code ausgeführt werden. Konnten alle Testfälle erfolgreich ausgeführt werden, d.h. die Systemantwort entspricht den in den Testfällen enthaltenen erwarteten Systemausgaben, so ist das System hinsichtlich der in den Testfällen enthaltenen Anforderungen verifiziert.



**Abbildung 5-2: Modellbasiertes Testen nach [LIG09]**

MBT unterstützt die soft-Inbetriebnahme, d.h. die Inbetriebnahme der Software ohne physikalische Anlage [HWO+10]. Dies wird auch als *Software-in-the-Loop* bezeichnet [VDI 2221]. Physikalische Komponenten können bei Bedarf zur Testfallausführung simuliert werden [HWO+10]. Wird die Automatisierungshardware in den Test mit einbezogen, so spricht man von *Hardware-in-the-Loop (HiL)* [VDI 2221]. Überprüft werden können sämtliche im Anforderungsmodell enthaltene Anforderungen, die meist funktionaler Natur sind, vgl. [LBL+14]. MBT wird auch genutzt, um sicherheitsgerichtete Funktionen zu testen [DIN EN 61508-3; FRKO15].

Konkret auf sich änderndes Verhalten zielt der sog. *Regressionstest* ab. Allgemein beschreibt dieser die Wiederholung einer Testausführung nach Änderungen, um sicherzustellen, dass die Änderung nicht das Fehlschlagen zuvor erfolgreicher Tests hervorruft [LIG09]. Diese Tests sind möglichst automatisiert durchzuführen, um zu garantieren, dass sich das System zur Testausführung stets im selben Ausgangszustand befindet und die Umgebungsbedingungen stabil sind [LIG09]. Insbesondere im Falle evolvierender Produktionssysteme werden Regressionstests als notwendig betrachtet, vgl. [RUP+15]. Einzelne Ansätze, z.B. [USV+14], beschäftigen sich mit der konsistenten Co-Evolution zwischen Testfällen und dem zu testenden System.

Alle Testverfahren haben gemeinsam, dass sie die Korrektheit der Software oder des Systems nicht beweisen können, sondern lediglich die Fehlerfreiheit in den durchgeführten Testszenarien [LIG09]. Ein Testverfahren, das alle möglichen Systemeingaben berücksichtigt, ist für reale Software i.d.R. nicht umsetzbar [LIG09]. Testmodelle müssen manuell erstellt werden, was komplex, aufwändig, annahmenbehaftet und fehleranfällig ist, vgl. [MKD15]. Zudem sind Testverfahren auf die Software und die Automatisierungshardware beschränkt. Entsprechend sind für die Testfallgenerierung Modelle notwendig, die einer adäquaten Beschreibung der Physik des Systems entsprechen [RUP+15]. Somit sind in allen Testfällen Annahmen über das korrekte Verhalten der Physik des Systems oder über bestimmtes vordefiniertes Fehlverhalten enthalten. Dass die Anlage nach Errichtung tatsächlich das in den Testfällen angenommene Verhalten aufweist, kann nicht garantiert werden.

Eine Ausnahme bildet hier der Ansatz von Ulewicz und Vogel-Heuser [UIVo16]. Der beschriebene Ansatz bezieht den Bediener in die Testausführung ein. Über ein Human-Machine-Interface kann der Bediener mit der Testfallausführung interagieren. Dies ermöglicht es entsprechend der Testfallergebnisse die Physik des Systems anzuregen und die resultierende Systemantwort zurück in die Testfallparametrierung und Ergebnisbewertung zu spiegeln. Somit können das reale physikalische Verhalten und auch Änderungen dessen berücksichtigt werden. Allerdings benötigt dieser Ansatz einen deutlich höheren manuellen Aufwand als konventionelle Test-Ansätze. Zudem muss das System, oder ein baugleiches Vergleichssystem, zur Verfügung stehen, um den Ansatz auszuführen. Insbesondere bei einzigartigen Fertigungssystemen, die möglichst kurze Stillstandzeiten aufweisen sollen, ist dies meist nicht der Fall.

### 5.1.2 Kritische Betrachtung

MBE dient dem systematischen Entwurf von Fertigungssystemen und/oder ihrer Automation. Ist ein korrektes Modell des Systems und evtl. der Anforderungen vorhanden, so ist eine zuverlässige, und ggf. automatisiert durchführbare Bewertung möglich. Die diskutierten Ansätze zeigen, dass eine Vielzahl an relevanten Systemeigenschaften zur Bewertung des Systems herangezogen werden kann.

Nachteilig ist, dass im Allgemeinen manuell modelliert wird und dieser Vorgang sich im Falle von Änderungen wiederholt [NiG14]. Ein Anlagenmodell inkl. des physikalischen Verhaltens ist bereits für die reine Softwarebewertung unabdingbar [MDL+06]. Ein Nachteil bei der Berücksichtigung des gesamten Verhaltens ist jedoch die daraus resultierende Modellkomplexität [VFS+15]. Des Weiteren werden hohe Ansprüche an die Modelle gestellt, für deren Erstellung ein Experte notwendig ist, der das Gesamtmodell sowie die geplante Änderung im Detail kennt, d.h. Wissen über den Prozess und alle Gewerke hat [NiG14]. Die Modellerstellung ist daher aufwendig und fehleranfällig [BFS10]. Einige Ansätze, wie bspw. [BuVy16], beschäftigen sich daher mit automatischer Modellgenerierung. Diese benötigen jedoch stets eine korrekte und aktuelle Spezifikation. Entwurfsmuster [Eck15] und Wiederverwendung [Sch16] sind zwei Ansätze, die den Aufwand in Entwurf, Modellierung und/oder Evaluierung reduzieren können. Aufgrund der Einzigartigkeit von Produktionssystemen [Fay09] ist jedoch stets eine systemspezifische Betrachtung notwendig. Zudem ist die Wahl der Modellgranularität für wiederverwendbare Modelle eine nicht-triviale Aufgabe [MJG11A; MJG11B]. Ein Ansatz, um mit Ungenauigkeiten in Modellparametern umzugehen, ist diese mithilfe von Laufzeitinformationen zu konkretisieren, vgl. [EGM+09]. Dieses kann zu einem besseren Abgleich zwischen Modell und realem System führen, reduziert den Modellierungsaufwand jedoch unwesentlich, da die Parameter im Modell bereits vorgesehen werden müssen. Eine Möglichkeit zur Reduzierung des Modellierungsaufwandes ist die Verwendung oder Ergänzung eines modelllernenden Ansatzes [VFS+15]. Diese werden im folgenden Abschnitt diskutiert (5.2).

Eine ähnlich komplexe Aufgabe ist die Formulierung der zu prüfenden Eigenschaften bzw. Anforderungen. Auch hier ist ein hoher Modellierungsaufwand nötig und die Modellierung ist entsprechend fehleranfällig [CAMA09]. Dies gilt für spezifizierte Eigenschaften z.B. in Form von temporalen Logiken wie für Testmodelle. Zudem sei zu beachten, dass Detaillierungsgrad und Modellierungsaufwand korrelieren [BAFA09]. Aufgrund der i.d.R. hohen Anzahl an Signalen ist eine detaillierte Modellierung meist nicht wirtschaftlich vertretbar [VDI 2221]. Um den Modellierungsaufwand moderat zu halten, muss also eine Abstraktion sowie eine Partitionierung und/oder Simplifizierung der Modelle stattfinden, was jedoch einer detaillierten Analyse des ganzheitlichen Verhaltens widerstrebt [Fay09]. Ergänzend kommt hinzu, dass die Planung der Systeme meist unvollständigen Informationen und Parameterschätzungen unterliegt [BRMA97; AIK+08] und häufig idealisierte Bedingungen angenommen werden. Einige Informationen sind nur implizit oder gar nicht vorhanden [BAFA09], andere sind reines Expertenwissen [GÜT13]. Dar-

über hinaus nennt Fay [FAY09] u.a. folgende Schwierigkeiten bei der Durchführung modellbasierter Vorgehensweisen:

- Individuelle Modellierung und Sichtweise des Modellierenden resultiert in individuellen Modellen, die schwer wartbar und weiterzuentwickeln sind
- Nachträgliche Änderungen müssen, um eine Diskrepanz zwischen Modell und Realisierung zu vermeiden, angepasst werden. Unter Beachtung der Qualifikation des Inbetriebnahme- und Betriebspersonals und der Zeitknappheit ist dieses Vorgehen nicht realistisch.
- Existierende Systeme können von den Vorteilen nicht profitieren, es müsste ein vollständiges Re-Engineering der Anlagen vorgenommen werden.

Insbesondere der zweite Punkt erschwert die Verwendung von MBE zur Evolutionsunterstützung. Da die Modelle auch der Dokumentation des Verhaltens dienen sollen, entsteht hier eine „Wissenslücke“ zwischen dokumentiertem und reellem Verhalten (siehe auch [LHF+14A<sup>#</sup>]). Dem Anspruch einer sich wandelnden Dokumentation, insbesondere des Systemverhaltens, konnten bisherige Ansätze bis heute nicht gerecht werden [BSK15].

Der Erfolg eines Produktionssystems hängt maßgeblich vom Übergang der Spezifikation zum realisierten System ab [NWW97]. Dabei ist die Übereinstimmung zwischen Modell und Realität das wichtigste Kriterium [LAN99]. Ist ein Systementwurf spezifiziert und erfolgreich bewertet, so muss das Verhalten des realisierten Systems auch dem der Spezifikation entsprechen. Aufgrund der hohen Diskrepanz zwischen idealisierter (Simulations-) Umgebung und den realen Bedingungen sowie eventuellen Fehlern in der Umsetzung des Modells zum System können Analyseergebnisse, die während des Entwurfs erreicht wurden, von den tatsächlichen Ergebnissen abweichen [HLW06]. Daher müsste das realisierte Gesamtsystem stets erneut bewertet werden [LIT09]. Dies wirkt dem eigentlichen Ziel der Aufwandsreduktion mittels automatischer Analyse wiederum entgegen.

## 5.2 Modellernende Ansätze

Auf Daten basierende modelllernende Ansätze entstammen ursprünglich dem *Machine Learning* und werden bereits seit den 1950er Jahren erforscht, vgl. [NER58]. Die hieraus erforschten Algorithmen, z.B. [BIFe72; GOL78; PoFe91; GJL04; MEL005; LAD+11], zeigen eine hohe Generalität und infolgedessen auch Komplexität, jedoch auch häufig unpraktische Nebenbedingungen auf [HNS03]. Domänenspezifische Algorithmen sind daher notwendig [HNS03] und einige sind bereits erforscht und entwickelt worden, vgl. [OPP02]. Im Rahmen technischer Systeme wird die Ermittlung von Modellen aus der Umsetzung der Systeme als *Reverse Engineering* bezeichnet [CHCR90]. Methoden hierfür können in statische und dynamische Methoden unterschieden werden [HUA05]. Die statischen Methoden bedienen sich einer Spezifikation oder des Quellcodes und generieren Modelle mittels Transformation, vgl. [FRLI00]. Statische Methoden, bspw. [NMS+13; RTF15], sind meist (Spezifikations- oder Implementierungs-) Sprachenabhängig und daher stark limitiert [HUA05].

Konträr zu den statischen Methoden bedienen sich die dynamischen Methoden an Beobachtungen des laufenden Systems. Einige Ansätze nutzen Daten, die aus Testausführungen der Software gewonnen werden können, bspw. [ECG+01; EGY03; ACH+10; TEL11; PSG14]. Andere Ansätze, bspw. [ABL02; LMP06; MPP11], nutzen interne Variablen, Parameter oder Interaktionssignale einzelner Komponenten der Software und deren Verläufe zur Generierung der Modelle. Diese Ansätze eignen sich jedoch nicht zur Bewertung des Gesamtsystems. Sie zeigen das Verhalten der Software, die korrektes Verhalten der weiteren Disziplinen voraussetzt oder fehlerhaftes Verhalten als Ausnahmen behandelt. Zudem kann ein

interner Zugriff auf die Software ihr Verhalten beeinflussen, oder er ist aus sicherheitstechnischen Gründen zur Anwendung in Fertigungssystemen nicht möglich. Andere Ansätze, bspw. [BDJ15], beschäftigen sich mit der Ermittlung kausaler Zusammenhänge aus Alarmlogs von Leitsystemen. Diese setzen jedoch ein Leitsystem voraus und benötigen zusätzlich internen Zugriff auf dieses, was für die vorliegende Arbeit ausgeschlossen wurde. Im Folgenden werden daher Ansätze diskutiert, die dynamische Methoden nutzen, um aus extern beobachtbarem Verhalten formale diskrete Modelle zu generieren.

### 5.2.1 Modellgenerierung im Rahmen der Fehlerdiagnose

Die Fehlerdiagnose umfasst nach [DIN 31051] „Maßnahmen zur Fehlererkennung, Fehlerortung und Ursachenfeststellung“. Dabei kann Diagnose in *signalbasierte*, *modellbasierte* und *wissensbasierte Diagnose* unterschieden werden [CHR15]. Die modellbasierte Diagnose nutzt Ein-/Ausgangsbeschreibungen des Systemverhaltens, um diese mit dem realen Verhalten des Systems zu vergleichen [ISE06]. Im Falle nicht übereinstimmenden Verhaltens von Modell und realem System aufgrund eines aufgetretenen Fehlers erlaubt eine Auswertung der Residuen die Bestimmung bzw. Abschätzung der Fehlerart und –Ursache [MAN04; RLL09B]. Die Erkennung nicht im Modell enthaltenden Verhaltens wird auch als *Anomalieerkennung* bezeichnet [NIL015]<sup>27</sup>. Da die Erstellung eines Verhaltensmodells zur Anomalieerkennung komplex, aufwändig und fehleranfällig ist [NIFR15] (vgl. Abschnitt 5.1.2), haben sich verschiedene Ansätze mit der automatisierten Modellerstellung auf Basis von Verhaltensbeobachtungen beschäftigt. Im Bereich der Produktion existieren viele Ansätze zur Modelgenerierung für kontinuierliche Systeme, vgl. [ISE06; ALAU13], sowie zur Erkennung von Änderungen im Signalverhalten, vgl. [BAN93]. Hybride Systeme werden bspw. im Ansatz von Niggemann et al. behandelt, z.B. [VKN+11; OBA+12; MAN15]. Ansätze der Fehlerdiagnose, die sich mit der Modellgenerierung und Anomalieerkennung für Discrete Event Systems (DES) beschäftigen, werden im Folgenden diskutiert.

#### Automatenbasierte Ansätze

Klein et al. [KLL05] präsentieren einen Ansatz zur Generierung von stochastischen Automaten zur Fehlererkennung. Hierzu werden mehrere I/O-Sequenzen des Systems aufgenommen und in Subsequenzen gleicher, vordefinierter Länge unterteilt. Diese Subsequenzen repräsentieren die Wörter der System-Sprache. Daher werden für jedes dieser Wörter ein Zustand erstellt und die Zustandsübergänge anhand der kausalen Reihenfolge der Wörter in den I/O-Sequenzen ermittelt. Daraufaufgehend finden einige Reduktionsschritte zur Modellkomprimierung statt. Da automatisierte Fertigungssysteme häufig eine hohe Anzahl an Nebenläufigkeiten aufweisen, welche in einer Zustandsexplosion und somit nicht überschaubaren Automaten resultieren, wurde der Ansatz in [RLL09A; RLL10] erweitert. Anstatt einen monolithischen Automaten zu generieren, werden die auftretenden Ereignisse partitioniert und für die entstehenden Untermengen je ein Automat gelernt. Ziel ist es, eine Menge an Automaten mit minimaler Nebenläufigkeit zu finden, was über einen Optimierungsalgorithmus gelöst wird. In einem zweiten Schritt werden dann gewisse Abhängigkeiten zwischen den Automaten gesucht. Dies kann zu beliebig abstrakten Modellen führen, die physikalisch (und evtl. auch logisch) nicht zusammenhängende Ereignisse vereinen, was die Analyse hinsichtlich nicht-funktionaler Eigenschaften auf den resultierenden Modellen kaum möglich macht. Allerdings ist dies durchaus ausreichend zur automatisierten Anomalieerkennung. Eine weitere Erweiterung in [RSL+12] ermöglicht es, aus unvollständiger Beobachtung der Sprache des Systems Modelle zu generieren. Der Algorithmus wird in [SLL12] nochmals erweitert, so dass auch die statistische Verteilung der Zeitabstände zwischen den Ereignissen annotiert werden kann. Die so gelernt-

---

<sup>27</sup> Allgemein muss die Anomalieerkennung nicht modellbasiert sein. Eine Übersicht zu Anomalieerkennungsalgorithmen für DES findet sich bspw. in [CBK12].

ten Automaten eignen sich zur Anomalieerkennung mittels zeitbehafteten Residuen [SLD11]. Somit ist es nicht nur möglich Anomalien in der kausalen Abfolge von Ereignissen zu erkennen sondern auch in den Zeitabständen zwischen den Ereignissen.

Einen ähnlichen Lernalgorithmus, der nicht direkt aus der Fehlerdiagnose stammt, jedoch aufgrund seiner Ähnlichkeit zu dem zuvor genannten hier erwähnt wird, präsentieren Verwer et al. [VWW08]. Der Algorithmus generiert zeitbehaftete Automaten aus einer Menge von abgeschlossenen Ereignissequenzen mit Zeitstempeln. Ziel des Algorithmus ist eine Minimalrealisierung des Automaten (eine Realisierung mit möglichst wenigen Zuständen), die in der Lage ist, die gegebenen Ereignissequenzen zu generieren. Um auch stochastische Zusammenhänge zu erfassen, wird der Algorithmus in [VER10] zur Generierung probabilistischer Automaten erweitert.

Wustmann et al. [SWV10; WVS+10] zeigen einen Ansatz auf, der sich mit der Diagnose von Materialflüssen beschäftigt. Ziel des Ansatzes ist es, sowohl Anomalien im Materialfluss als auch eine Bewertung in Bezug auf Leistungskennzahlen vornehmen zu können. Sie gehen dabei von einem Ereignislog des Materialflusssystemes aus, der für jedes Ereignis den Ort und Zeitstempel des Ereignisses enthält sowie die ID des Ereignis auslösenden Fördergutes. Hieraus generieren Sie einen automatenähnlichen Graphen, in dem jeder Knoten einen Ort und jede Kante einen Transportweg zwischen den Orten darstellt. Zudem enthält das Modell Pfade als Aneinanderreihung von Kanten, die Fördergüter im System zurücklegen können. Dieses wird zur Analyse in einen zeitbehafteten Automaten überführt. Der Ansatz zeigt, dass es möglich ist, auf Basis von Ereignisdaten eine Kenngrößen-basierte Analyse des Materialflusses vorzunehmen. Allerdings gehen die Algorithmen von Ereignissen aus, die bereits den Ort und die Stückgut-Instanz, also eine eindeutige Zuweisung zu einem konkreten Werkstück, mitliefern. Diese Informationen sind in SPS-I/Os nicht vorhanden. Der Ansatz ist also nur anwendbar, wenn die genannten Informationen zur Verfügung stehen. Es bleibt die Frage offen, ob eine entsprechende Analyse lediglich auf Basis von I/O-Ereignissen möglich ist und welche Zusatzinformationen vonnöten sind.

### **Petri-Netz-basierte Ansätze**

Cabasio [CAB09] beschäftigt sich mit der Identifikation und Diagnose von DES mittels Petri-Netzen. Hierbei geht sie davon aus, dass die Anzahl der Stellen ebenfalls bekannt oder näherungsweise bekannt ist, und konzentriert sich auf die Identifikation der Netzstruktur unter der Annahme, dass die gesamte Sprache des Systems bekannt ist (vgl. auch [CGS07]). Hierzu nutzt sie einen integer linear programming Algorithmus. Ein ähnlicher Ansatz findet sich in [DFM07]. Dieser setzt jedoch keine Kenntnis der gesamten Sprache voraus, sondern nimmt ein Modellupdate mit der Beobachtung jedes neuen Verhaltens vor. Der Algorithmus wird in [DFM+08] dahingehend erweitert, dass auch nicht beobachtbare Ereignisse identifiziert und in Petri-Netz-Fragmenten modelliert werden können. Als nicht beobachtbar werden diejenigen Ereignisse bezeichnet, die nicht direkt in den I/Os reflektiert sind. Nicht beobachtbare Ereignisse entsprechen also internen Ereignissen des Systems. Der Ansatz wird von Saives et al. [SFL15] aufgenommen und dahingehend erweitert, dass das Petri-Netz auch aus einer Sequenz von Transitions-Feuervorgängen heraus erlernt werden kann, indem die Petri-Netz-Fragmente zu einem Netz kombiniert werden.

Estrada-Vargas et al. präsentieren in [ELL12] einen Ansatz zur Identifikation von interpretierbaren Petri-Netzen aus der Beobachtung von SPS-I/Os gesteuerter DES wie diskreten Fertigungssystemen (siehe auch [ELL14]). Hierzu werden die I/Os zu einem Vektor zusammengefasst und es wird davon ausgegangen, dass Sequenzen dieses Vektors beobachtet wurden. Der Algorithmus berechnet die Ereignisse innerhalb der Sequenzen, die in gleichlange Subsequenzen unterteilt werden. Anschließend werden für jede Subsequenz Stellen-Transitions-Ketten mit annotierten Ereignissen erstellt. Gleiche Subsequenzen

werden zusammengefasst und Zyklen identifiziert. Anschließend werden Simplifikationen an dem Modell vorgenommen. Dieser Algorithmus wird in [ELL15] nochmals erweitert. Der Algorithmus liefert nach Aussage der Autoren Petri-Netze, die die kausalen Zusammenhänge des beobachteten Verhaltens wiedergeben. Ein ähnlicher Algorithmus, der aus einer Ereignissequenz T-Invarianten eines Petri-Netzes berechnet, wird von Tapia-Flores et al. [TLE+14] präsentiert. Aus den T-Invarianten kann entsprechend das Petri-Netz-Modell abgeleitet werden. Im Anschluss werden einige Zusammenführungen und Simplifikationen vorgenommen sowie die Initialmarkierung bestimmt. Weitere ähnliche Algorithmen zur Identifikation von (interpretierten) Petri-Netzen aus gegebenen Ereignissequenzen oder der Sprache des Systems finden sich in [FAS08]. Diese unterscheiden sich von den hier genannten lediglich in ihrer Performance und der zugrundeliegenden Methode. Gemein haben diese Ansätze, dass sie nicht das zeitliche Verhalten des Systems berücksichtigen. Für eine Analyse der Modelle hinsichtlich nicht-funktionaler Eigenschaften des Systems sind Zeitinformationen jedoch zwingend notwendig (vgl. Kapitel 3).

Einen Ansatz, der sich mit der Identifikation von zeitbehafteten Petri-Netzen beschäftigt, wurde von Lefebvre und Leclercq erarbeitet [LELE11]. Der dort präsentierte Lernalgorithmus basiert auf einer Kausalitätsspezifikation zur Identifikation der Petri-Netze aus den Vorgänger-Nachfolger-Beziehungen der Ereignisse in der Sequenz. Das Zeitverhalten wird in Form von Feuerfrequenzen der mit den Ereignissen annotierten Transitionen dargestellt. Zusätzlich diskutieren Lefebvre und Leclercq den für die Anomalieerkennung wichtigen Punkt der Permissivität. Permissivität bezieht sich auf die Eigenschaft des Modells, mehr Ereignissequenzen zu produzieren als die zum Lernen verwendete Sequenz. Dies wird in der Literatur auch als *Overfitting* bezeichnet [ISE06]. Im Kontext der Anomalieerkennung reflektieren maximal permissive Modelle sämtliche Ereignissequenzen, die mit den gegebenen Symbolen (Ereignissen) möglich sind, und würden keine Anomalie erkennen. Im Gegensatz dazu erlauben minimal permissive Modelle nur exakt die Ereignissequenz, aus der das Modell erlernt wurde. Entsprechend benennen Lefebvre und Leclercq die Kontroverse, ein generell gültiges Modell generieren zu wollen, das jedoch möglichst wenig permissiv ist.

### **Ansätze unter Berücksichtigung von a-priori-Informationen**

Dass es möglich ist, mithilfe von Lernalgorithmen Modelle zu lernen, die es erlauben Analysen auf Basis höherwertiger Eigenschaften bzw. Kenngrößen vorzunehmen, zeigt der Ansatz von Wustmann et al. [WVS+10] (s.o.). Hierbei gehen sie von einem Ereignis-Log aus, das neben dem Auftreten des Ereignisses und dessen Zeitpunkt weitere Informationen enthält. Somit ist es möglich die Analyse auf ein höheres Abstraktionsniveau zu heben. Steht, wie in dieser Arbeit angenommen, lediglich das Ereignis und dessen Auftrittszeitpunkt zur Verfügung, so können Informationen zu den Ereignissen, die schon vorab bekannt gemacht wurden, ebenfalls zur Anhebung der Analyseabstraktion beitragen. Entsprechende Ansätze werden in den folgenden Absätzen diskutiert.

Ein Ansatz, der a priori Informationen in Kombination mit einem Lernalgorithmus für Petri-Netze verwendet, wird von Allen [ALL10] vorgestellt, siehe auch [ALT10; ALT11]. Der Ansatz geht von DES-Prozessen aus, die auf gleiche Ressourcen zugreifen ("Systems of Processes with Shared Resources") und generiert die Modelle aus den in den Automatisierungssteuerungen (z.B. SPS) auftretenden Ereignissen. Zur besseren Interpretation der Modelle bzw. auftretender Fehler sei vorab bekannt, welche Ereignisse welchen Prozessen und Ressourcen zugeordnet sind und welche Ereignisse Ressourcen für Prozesse reservieren und wieder freigeben. Als Lernalgorithmus wird ein angepasster Process Mining Algorithmus verwendet (siehe Abschnitt 5.2.2). Die entstehenden Petri-Netze eignen sich zur Anomalieerkennung [BAT11; ALT12]. Die vorab-Information zur Semantik der Ereignisse erlaubt es zum einen Modelle zu erlernen, die besser verständlich sind, und zum anderen entdeckte Anomalien besser zu

interpretieren und evtl. Gegenmaßnahmen zu ergreifen. Nachteilig an diesem Ansatz ist, dass nicht garantiert ist, dass die für die Algorithmen benötigten Ereignisse vorhanden sind. Außerdem ist fraglich, ob, wenn vorhanden, die Ereignisse extern beobachtbar sind, oder ein interner Zugriff notwendig ist, um die Ereignisse zu beobachten.

Huselius et al. [HAH+06] präsentieren einen ähnlichen Ansatz zur automatischen Modell-Generierung evolvierender Systeme. Sie gehen davon aus, dass das beobachtete System Aufgaben (Tasks) erfüllen kann, die Ereignis- oder Zeit-getriggert werden. Wird eine Aufgabe angefordert, so wird diese in Form eines Jobs für eine gewisse Zeit ausgeführt. Dabei werden die folgenden Ereignisse aufgenommen und zur späteren Modellgenerierung genutzt:

- Senden und Empfangen von Daten-Operationen
- Variablen-Updates
- Kontext-Änderungen

Die Ereignisse werden an der Schnittstelle des Systems gesammelt und mit bestimmten Aktionen und zugehörigen Parametern assoziiert. Die Daten werden aus der Software zur Laufzeit gewonnen [HUA05]. Der in [HUA05] beschriebene Algorithmus zur Modell-Generierung separiert die Sequenz zunächst nach den Tasks, die sich aus den Aktions-Parametern ermitteln lassen. Aus den aufbereiteten Daten werden dann sog. Task-Trees gebildet. Jedes Blatt (Knoten, auf den kein weiterer Knoten folgt) eines solchen Baumes entspricht dabei einer Aktion, die mitsamt ihrer Parameter beschrieben ist. Zudem werden im Falle mehrerer möglicher Ereignis-Pfade Eintrittswahrscheinlichkeiten angegeben. Die Task-Trees werden dann in eine textuelle Beschreibung und zur Validierung gem. [HAH+06] in zeitbehaftete Automaten transformiert. Des Weiteren werden zusätzliche Daten des Systems herangezogen und ebenfalls Automaten in Form von Zustandsketten daraus generiert. Mittels Model-Checking wird ermittelt, ob der Endzustand einer jeden Zustandskette von den zuvor generierten zeitbehafteten Automaten erreicht werden kann. Die so gewonnenen Modelle dienen der Dokumentation und können zur Simulation herangezogen werden, wie in [HAH+06] aufgezeigt. Als Ergebnis einer Analyse werden Ausführungszeiten der Tasks ermittelt. Allerdings geht auch dieser Ansatz von bestimmten abstrakten Ereignissen aus, die im System messbar sein müssen und i.d.R. nicht in SPS-I/Os reflektiert sind. Zudem müssen diese in Form von Aktionen, die mit dem o.g. Parametersatz beschrieben sein müssen, verfügbar sein. Dies fordert entweder, dass die Software einem bestimmten Programmieransatz folgt, oder die Aktionen müssen in die Software hinein interpretiert und generiert werden. Eine Analyse hinsichtlich typischer nicht-funktionaler Anforderungen ist nicht vorhanden. Die Analyse beschränkt sich hier auf die Ausführungszeiten.

### 5.2.2 Process Mining

Ein Wissenschaftsgebiet, das sich ebenfalls mit der Generierung von Verhaltensmodellen, hier für Geschäftsprozesse, aus Ereignisdaten beschäftigt, ist das *Process Mining* [VABE03; VAWE04; VAN06; VAN11; VAN12]. Aufgrund der gleichen Zielsetzung wird dieses hier ebenfalls diskutiert. Das auch als *Workflow Mining* bezeichnete (vgl. [ELS10]) Gebiet lässt sich in die drei Teilbereiche *Process Discovery*, *Conformance Checking* und *Model Enhancement* unterteilen [VAN11]. Ersteres bezieht sich auf das Lernen von Modellen, meist Petri-Netzen, aus Ereignis-Daten und zweites auf die Anomalieerkennung mittels Modell- und Datenabgleich. Model Enhancement entspricht der Modellerweiterung auf Basis neu gewonnener Daten. Ziel ist die Diagnose von Geschäftsprozessen. Hierzu werden Ereignis-Logs herangezogen, die typischerweise in Informationssystemen wie SAP o.ä. generiert werden. Die Logs enthalten entsprechend für jedes Ereignis mindestens eine ID, einen zugehörigen Prozess (Case), einen Zeitstempel und

eine zugehörige Aktivität, vgl. [VAN16<sup>®</sup>]. Process Mining hat sich mittlerweile als eigenes Forschungsgebiet etabliert und es existiert eine Vielzahl an wissenschaftlichen Publikationen und Algorithmen. Als Beispiele seien der Algorithmus von Esparza et al. [ELS10] zur Generierung von Petri-Netzen aus unvollständigen Logs und von Pinter und Golani [PIGo04] zur Generierung von Modellen aus Aktivitätsdauern genannt. Aufgrund der Vielzahl an verfügbaren Algorithmen finden insbesondere Process Discovery Algorithmen auch Anwendung in anderen Domänen (siehe auch Abschnitt 5.2.1).

Der Ansatz von Cook und Wolf [CoWo98] ist ein Beispiel aus dem Software-Engineering. Cook und Wolf nutzen Ereignis-Daten, die im Laufe eines Software-Entwicklungsprozesses generiert werden, um diese in Form von endlichen statistischen Automaten darzustellen. Hierzu untersuchten Sie unterschiedliche Process Discovery Algorithmen.

Aus der Domäne der Produktion haben Hu et al. [HLW06], angelehnt an das Process Mining, einen Process Discovery Ansatz zur Generierung von Petri-Netzen für flexible Fertigungssysteme erarbeitet. Jeder Eintrag des hierfür zugrundeliegenden Ereignis-Logs beinhaltet die Aktivität, die das Ereignis auslöste, die Zuordnung zu einem Prozess und zu einer Ressource sowie Start- und Endzeit des Ereignisses. Ähnlich zu dem Ansatz von Allen (siehe Abschnitt 5.2.1) wird ein Petri-Netz gewonnen, das die Zuordnung von Ressourcen zu Aktivitäten sowie die kausale Abfolge von Aktivitäten innerhalb der Prozesse beschreibt.

### 5.2.3 Kritische Betrachtung

Algorithmen zur Generierung von Modellen aus Beobachtungen des Verhaltens von Discrete Event Systems existieren in hoher Anzahl, wie vorangegangene Diskussion gezeigt hat. Der wesentliche Vorteil dieser Ansätze liegt in der Reduzierung des Modellierungsaufwandes, da kein oder kaum Aufwand für die Erstellung der Modelle notwendig ist. Zudem ist garantiert, dass die Modelle auch dem tatsächlichen Verhalten entsprechen, da keine manuellen Modellierungsfehler vorkommen können. Dies gilt allerdings nur für das beobachtete Verhalten. Ob das Verhalten vollständig beobachtet wurde, ist damit nicht garantiert. Zudem ist es notwendig, das Verhalten fehlerfrei zu beobachten.

Viele Ansätze entstammen der Fehlerdiagnose und dienen dem Abgleich von erlerntem Verhalten mit Beobachtungen des aktuellen Systemverhaltens (Anomalieerkennung). Im Falle einer Anomalie können Diagnose-Ansätze, z.B. Bildung von Residuen, Hinweise auf die Fehlerursache geben, indem sie die Anomalie eingrenzen. Dies kann auch zur Erkennung von gewollten Änderungen im System dienlich sein. Eine Analyse auf einem höheren Abstraktionsniveau, wie nicht-funktionalen Eigenschaften des Produktionsprozesses, lassen die Modelle allerdings nicht zu. Dies liegt an dem geringen Abstraktionsgrad der gelernten Modelle. Auch der Dokumentationscharakter ist aufgrund des geringen Abstraktionsniveaus schwach ausgeprägt. Eine Interpretation von Daten sowie Anomalien in den Daten auf einer höheren Abstraktionsebene ist nur möglich, wenn die Daten mit einer Semantik versehen sind, vgl. [SPK15]. Einige Ansätze nutzen neben dem Auftreten und Zeitpunkt des Ereignisses statische Informationen, die vorab bekannt sind, um das Abstraktionsniveau der Modelle bzw. deren Analysierbarkeit anzuheben [HUA05; HLW06; ALL10; WVS+10]. Diese Ansätze gehen davon aus, dass gewisse Ereignisse vorhanden sind oder die Ereignisse bereits Prozessinstanzen und/oder Ressourcen zugewiesen sind. Gleiches gilt für Process Discovery Algorithmen, die einer Zuordnung der Ereignisse zu Prozessen und Aktivitäten bedürfen.

Allerdings kann das Vorhandensein spezieller Ereignisse oder der Zuordnung von Ereignissen zu Prozessen und/oder Ressourcen nicht allgemeingültig angenommen werden. Dies zeigt sich bereits darin, dass jeder der o.g. Ansätze von unterschiedlichen Ereignissen oder Zuweisungen ausgeht. Eine Fallstudie in

[VRF+16] hat ähnliches gezeigt. Ein Ergebnis der Fallstudie ist, dass viele Ereignisse, die für eine Modellierung und Analyse auf einem höheren Abstraktionsniveau (hier am Beispiel eines Fehlerbaumes) in realen Systemen nicht vorhanden sind. Folglich reduzieren lernende Ansätze den Modellierungsaufwand zwar drastisch, lassen jedoch keine Analyse auf einem höheren Abstraktionsniveau und insbesondere hinsichtlich typischer nicht-funktionaler Eigenschaften zu.

### 5.3 Industrielle Praxis

Aktuelle in der Industrie genutzte Engineering-Prozesse weichen häufig von den in Standards und Forschung empfohlenen ab, insbesondere in der Erstellung von expliziten (formalen) Modellen [WIT12]. Dies gilt auch für die Durchführung von Änderungen, wie bspw. eine Studie mit 100 Unternehmen der Fertigungsindustrie in [HUMA99] zeigt. Eine deutliche Diskrepanz findet sich bereits im Requirements-Engineering, was u.a. zu mangelhaft dokumentierten und/oder nicht implementierbaren Anforderungen führt [BGK+07]. Als Gründe hierfür nennt Morris fehlendes Know-How aufgrund mangelnder Ausbildung, die Komplexität aufgrund der Vielzahl an Stakeholdern und Anforderungskonflikten, fehlende Integration in die Geschäftsprozesse sowie eine mangelnde Akzeptanz durch das Management [MMW98]. Zudem ergeben sich aus der Forderung nach kurzen Stillstandzeiten ungenügende Rahmenbedingungen für die Durchführung eines systematischen Anforderungsmanagements [VFS+15]. Dem zugrunde liegt häufig die Forderung nach einer schnellen Realisierung eines produktiven Systems [PoRU11]. Zeit- und Kostendruck entstehen daher nicht zuletzt aufgrund des Strebens nach möglichst kurzer Time-to-Market [BESÄ10]. Folglich werden Anforderungsspezifikationen in der Produktionsentwicklung nicht immer erstellt [BESÄ10]. Geschieht dies doch, werden häufig Fehler begangen, die erst in späten Phasen wie der Endabnahme erkannt werden [PoRU11]. Fehlerhafte Anforderungen werden beschrieben, da häufig gewisse Sachverhalte als selbstverständlich oder allgemeingültig angesehen werden [PoRU11]. Dies bestätigt auch eine Studie zur Produktionsentwicklung in [BESÄ10].

Des Weiteren mangelt es häufig an der Formalisierung der Anforderungen [FRLI00]. Dies gilt insbesondere für kurzfristige Änderungen [VFS+15]. Studien zeigen, dass im Falle kleiner Änderungen Anforderungen oft nur verbal kommuniziert werden [BESÄ10]. Anforderungsspezifikationen werden insbesondere dann unterschätzt, wenn sie für den internen Gebrauch gedacht sind und nicht wertvolles mechanisches Equipment betreffen [BESÄ10]. Folglich werden informelle Anforderungen direkt implementiert ohne formale Zwischenschritte [FRLI00]. Häufig liegen Anforderungen auch in unterschiedlichen Formalisierungsgraden vor und Ingenieure sehen sich mit einer Vielzahl heterogener Dokumente konfrontiert [STFA11]. Während des Anlagenengineerings wird häufig funktionsorientiert gearbeitet. Dies hat zur Folge, dass NFAs gar nicht oder zu spät berücksichtigt werden [ORG+11; Eck15]. Des Weiteren können funktionale Anforderungen meist in Basis-Funktionen herunter gebrochen werden, während die NFAs eher vom emergenten Verhalten des gesamten Systems abhängig sind [VFS+15]. Deshalb werden insbesondere Qualitätsanforderungen, wenn überhaupt, lediglich in natürlicher Sprache formuliert [PoRU11]. Der Umgang mit Anforderungen und formalen Spezifikationen unterscheidet sich hier stark von der Vorgehensweise in der Produktentwicklung [BESÄ10], denn Änderungs- und Anforderungsmanagement von individuellen Lösungen, wie Produktionssystemen, gestaltet sich schwieriger als für Standardlösungen [VDMc14]. Häufig steht ohnehin die Produktentwicklung im Fokus und die Entwicklung des Produktionssystems rückt in den Hintergrund, wie auch Studien belegen [BESÄ10]. Dies führt dazu, dass insbesondere Änderungen der Produktion, die nicht aus geänderten Produkten resultieren, ohne Änderungsmanagement durchgeführt werden [JMR04]. Folglich werden eher Standardlösungen verwendet, ohne eine explizite Analyse hinsichtlich ihrer Auswirkungen auf das spezifische System. Dies führt dazu, dass „Copy, Paste and Modify“ eine häufig angewandte Methode im Engineering von Produktionssystemen und

deren Änderungen ist [MJG11b]. Entscheidungen werden oft aus Projektdiskussionen heraus und intuitiv auf Basis von Erfahrungswissen getroffen [CHR06; BESÄ10]. Entsprechend weisen Modelle sämtliche Formalisierungsgrade auf. Insbesondere die Modelle der Anforderungen und des zu automatisierenden Systems sind häufig hochgradig informell [VDF+14]. Eine Studie in [VRF+16] zeigt, dass existierende Anlagen MBE-Ansätze schwer verfolgen können, da bestehende Automatisierungslösungen und Implementierungen nicht hierauf ausgerichtet sind und folglich ein Neuentwurf der AT-Software notwendig wäre. Dies ist entsprechend aufwändig und somit zeit- und kostenintensiv. Selbst wenn Anlagen mithilfe formaler Modelle entworfen wurden, werden besonders die Pflege von Systemmodellen mit viel Aufwand verbunden und die Modelle meist nicht über die gesamte Lebensdauer gepflegt [HAH+06]. Im Falle von Änderungen wird also eine Anpassung der Spezifikation bzw. Dokumentation unzulänglich oder gar nicht vorgenommen [FEVO13]. Dies führt dazu, dass Erfahrungswissen meist nur in Planungsergebnissen oder dem System selbst resultiert, jedoch häufig nicht explizit verfügbar ist und dann für Folgeplanungen o.ä. wiederbeschafft werden muss [SCH08; VFS+15].

Eine Bewertung des Systems im Anschluss an durchgeführte und in Betrieb genommene Änderungen wird ebenso häufig vernachlässigt [HUMA99; BESÄ10; LWM+12]. Selbst wenn eine (formale) Anforderungsspezifikation vorliegt, wird eine auf Methodik basierende systematische und dokumentierte Evaluierung von Systemlösungen häufig ausgelassen [BESÄ10; LWM+12]. Insbesondere Änderungen, die nicht als Projekt durchgeführt werden (also kleine, ad-hoc-Änderungen, die Teil der täglichen Arbeit sind), werden hinterher nicht evaluiert [BESÄ10]. Dabei können auch kleinere Änderungen, z.B. in Steuerungssystemen, gravierende Folgen haben [VDI 2221]. Die Durchführung von Prozessanalysen ist in der Praxis jedoch sehr aufwändig, was nicht zuletzt an den komplexen Interdependenzen und der Vielzahl an zu erhebenden Prozessmerkmalen liegt [AIK+08]. Taylor bemängelt, dass die Erhebung vieler Metriken schwer erhebbare Daten benötigt [TAY97]. Somit kann als Grund für eine fehlende Evaluierung auch genannt werden, dass eine immer schnellere Implementierung bei immer höherer Komplexität der Anlagen gefordert wird (vgl. [ROG11]). Zudem sind Formalisierungs- und Evaluierungsprozesse mit teils erheblichen Kosten verbunden [BESÄ10; SDK+11]. Auch die Qualifikation des Personals spielt hier eine Rolle. Häufig ist Anlagenpersonal, welches i.d.R. für die Durchführung kleinerer Änderungen zuständig ist [VDF+14], nicht entsprechend ausgebildet, um mit formalen Modellen und Analysen zu arbeiten [FRLI00; ScFA13]. Das qualifizierte Personal, das für das Requirements-Engineering und/oder die Modellierung während der Planung der Anlage zuständig war, ist im operativen Betrieb und damit während der Evolution der Anlagen meist nicht mehr verfügbar [VFS+15].

Es lässt sich somit schlussfolgern, dass zwar viele Methoden, die es ermöglichen Anlagenevolution adäquat zu unterstützen, vorhanden sind, diese jedoch in der Praxis derzeit keine Anwendung finden. Die Hauptgründe liegen in dem zeitlichen Aufwand und dem Bedarf an qualifiziertem Personal zur Durchführung der Ansätze.

## 6 Forschungsbedarf, Anforderungen und These

Um Änderungen in gewünschter Qualität vornehmen zu können, sollten alle vorangegangenen Änderungen sowie die ursprüngliche Implementierung ausreichend bekannt und dokumentiert sein [ALEL11]. Aufgrund der Interdisziplinarität und der daraus resultierenden Komplexität der Systeme ist es wahrscheinlich, dass durchgeführte Änderungen zu unvorhergesehenen und ungewollten Verhaltensänderungen führen [JFW+11; Wit12]. Um der daraus resultierenden Degenerierung vorzubeugen, ist es unabdingbar eine konstante Bewertung des Systems mittels Beurteilung seiner Merkmalsausprägungen vorzunehmen. Für evolvierende Systeme ist es entsprechend notwendig, vorhandene Modelle koordiniert mit zu evolvieren [KMO+10; SDK+11]. Die Co-Evolution zwischen Modellen und der Anlagenrealisierung gehören daher zu den Kern-Herausforderungen in der Forschung zu evolvierenden automatisierten Produktionssystemen, genauso wie die Qualitätssicherung und die Qualitätsüberprüfung [VFS+15; VFF+15]. Aus der mangelnden Anwendung bisheriger Ansätze (siehe Abschnitt 5.2.3) ergibt sich der Bedarf an einer neuen evolutionsunterstützenden Methode. Dies gilt insbesondere für Änderungen, die in Tabelle 2-1 (Abschnitt 2.3.3) in die Kategorien III – IV fallen. Dies sind diejenigen Evolutionsszenarien, in denen keine Anpassung der Systemspezifikation vorgenommen wird. Somit mangelt es ihnen auch an geeigneter Bewertung. Dieses Kapitel stellt zunächst die Anforderungen an einen solchen Ansatz zusammen, diskutiert explizit die Forschungslücke, die sich aus bisherigen Ansätzen ergibt und formuliert schlussendlich eine These, die es in dieser Arbeit zu belegen bzw. widerlegen gilt.

Um eine Änderung zu bewerten, ist es zunächst notwendig diese zu erkennen. Dies gilt insbesondere auch für o.g. Seiteneffekte, die evtl. ungewollte Verhaltensänderungen hervorrufen. In Bezug auf das Verhalten des Systems vor der Änderung, entspricht geändertes Verhalten einer Anomalie. Daher kann das Feststellen einer Evolution als *Anomalieerkennung* bezeichnet werden. Ist eine Anomalie erkannt und erfasst, muss eine Analyse des geänderten Verhaltens zur Bewertung vorgenommen werden, die auch den Fokus dieser Arbeit darstellt. Daraus ergibt sich die folgende Anforderung:

### Anforderung 6-1

Anlagenevolution muss vollständig erkannt und bewertet werden. Entsprechend sind eine Anomalieerkennung und eine Analyse geänderten Verhaltens notwendig.

Insbesondere sind die Anforderungen, die sich aus den Rahmenbedingungen der industriellen Praxis ableiten lassen, einzuhalten. Die wichtigste Anforderung ergibt sich daraus, dass Aufwände gering zu halten sind. Wie Abschnitt 5.2.3 zeigt, werden viele erfolgversprechende Ansätze nicht angewandt, da sie mit einem hohen Modellierungs- und entsprechendem Zeit- und Kostenaufwand verbunden sind oder aus Mangel an Informationen eine Modellierung gar nicht möglich ist:

### Anforderung 6-2

Manueller Aufwand sollte möglichst gering gehalten werden. Die Modellierung notwendiger Informationen sollte einfach gehalten werden und auf Informationen zurückgreifen, die in jeder Anlage verfügbar sind.

Gleichzeitig ist es notwendig, das zu untersuchende System möglichst wenig zu beeinflussen [KMO+10]. Zum einen muss verhindert werden, dass der produktive Betrieb durch Anwendung der gesuchten Me-

thode gestört wird. Zum anderen soll auf einen Zugriff der Steuerungssoftware verzichtet werden, um Einsicht in sensibles Know-how in Form von Steuerungscode zu vermeiden und somit die Akzeptanz für einen entsprechenden Ansatz zu erhalten. Daraus resultiert, dass lediglich extern bzw. an den Schnittstellen des Systems verfügbare Daten verwendet werden sollen:

**Anforderung 6-3**

Dynamische Informationen sollten extern verfügbar sein und die Beschaffung der Daten sollte weder die Ausführung des Fertigungsprozesses beeinflussen, noch Zugriff auf Quellcode oder bestimmte Entwicklungstools benötigen.

Wichtig ist, dass Analyseergebnisse durch das Anlagenpersonal auch interpretierbar sind, idealerweise auch durch das Anlagenmanagement, welches die Anforderungen festlegt. Eine Analyse hinsichtlich typischer nicht-funktionaler Anforderungen, wie sie in Abschnitt 3.3 diskutiert wurden, ist daher wünschenswert. Dass sich dieses Problem in die Messung einzelner Qualitätsmerkmale mittels einer geeigneten Messmethode dekomponieren lässt, wurde in Abschnitt 3.2 diskutiert:

**Anforderung 6-4**

Die Analyse muss auf einem Abstraktionsniveau geschehen, welches typische nicht-funktionale Anforderungen abdeckt. D.h. es sollten Merkmalsausprägungen ermittelt werden, die entsprechend der Literaturrecherche in Kapitel 3 relevant sind. Eine möglichst große Untermenge der Merkmale sollte von dem Ansatz abgedeckt werden.

Kapitel 5 zeigt, dass modellbasierte Ansätze Analyseergebnisse auf dem gewünschten Abstraktionsniveau liefern können (Anforderung 6-4). Die erstellten Modelle stellen eine gute Dokumentation des Systems dar und sie eignen sich sehr gut zur Systemanalyse. Dies erfüllt einen Teil von Anforderung 6-1. Dem gegenüber stehen allerdings der hohe manuelle Modellierungsaufwand und der hohe Anspruch an den Modellierer. Dies widerspricht Anforderung 6-2. Modelllernende Ansätze hingegen reduzieren den Modellierungsaufwand erheblich. Zudem sichern sie zu, dass das modellierte Verhalten auch dem tatsächlichen entspricht, da es aus Beobachtungen dessen generiert wurde. Viele Ansätze verwenden hierbei extern verfügbare I/O-Signale (Anforderung 6-3). Ansätze der Fehlerdiagnose haben zudem gezeigt, dass mithilfe generierter Modelle geändertes Verhalten erkannt werden kann. Dies kommt Anforderung 6-1 zu Gute. Nachteilig ist allerdings, dass der Abstraktionsgrad der generierten Modelle gering ist (Abschnitt 5.2).

Tabelle 6-1 fasst alle in Abschnitt 5.1 diskutierten Vor- und Nachteile in einer SWOT<sup>28</sup>-Analyse zusammen. Zudem sind die Vor- und Nachteile der modelllernenden Ansätze in einer SWOT-Analyse in Tabelle 6-2 zusammengefasst.

<sup>28</sup> Strengths, Weaknesses, Opportunities, Threats

**Tabelle 6-1: SWOT-Analyse zu modellbasierten Verfahren**

<u>Stärken</u>	<u>Schwächen</u>
<ul style="list-style-type: none"> <li>- Analyseergebnisse zu abstrakten und relevanten Eigenschaften/Anforderungen</li> <li>- Modelle besitzen starken Dokumentationscharakter</li> </ul>	<ul style="list-style-type: none"> <li>- Hoher Modellierungsaufwand</li> <li>- Komplexe Modellierung muss durch entsprechend geschultes Personal durchgeführt werden</li> <li>- Fehleranfällig</li> </ul>
<u>Möglichkeiten</u>	<u>Bedrohungen</u>
<ul style="list-style-type: none"> <li>- Co-Evolution von Modell und Anlage</li> <li>- Hinzunahme von Laufzeitinformationen</li> <li>- Codegenerierung möglich</li> </ul>	<ul style="list-style-type: none"> <li>- Nicht-Aktualisierung der Modelle während der Systemevolution</li> <li>- Realisierung des Systems entspricht nicht dem Modell aufgrund fehlerhafter Umsetzung oder Abstraktion</li> </ul>

Auffällig ist, dass Stärken und Schwächen der beiden Ansätze komplementär zueinander sind. Insbesondere verlangen modellbasierte Ansätze einen hohen Modellierungsaufwand, während dies bei modelllernenden Verfahren nicht der Fall ist. Im Gegensatz dazu liefern modellbasierte Ansätze verständliche Modelle, die einen hohen Dokumentationscharakter aufweisen, während Modelle, die mit modelllernenden Verfahren gewonnen werden, einen sehr geringen Abstraktionsgrad aufweisen. Es liegt also nahe, einen Ansatz zu verfolgen, der modelllernende Algorithmen verwendet, um Modelle auf einem Abstraktionsniveau zu generieren, das typischerweise in modellbasierten Methoden vorzufinden ist. Einige Ansätze zeigen, dass Interpretationen von automatisch generierten Modellen auf einem höheren Abstraktionsniveau möglich sind, wenn statische a priori Informationen hinzugezogen werden [HAH+06; ALL10; WVS+10]. Die bisherigen Ansätze berufen sich jedoch auf die Existenz spezieller Ereignisse, von denen nicht im Allgemeinen ausgegangen werden kann. Vielmehr muss eine generelle Methode gefunden werden, die es ermöglicht die Qualität eines Systems mittels einer Qualitätsmessmethode (vgl. Abbildung 3-3) unter Zuhilfenahme von Lernalgorithmen zu ermitteln. Da gem. Anforderung 6-3 nur extern verfügbare Daten zur Verfügung stehen, sollten lediglich I/O-Signale als dynamische Informationen herangezogen werden.

**Tabelle 6-2: SWOT-Analyse zu modelllernenden Verfahren**

<u>Stärken</u>	<u>Schwächen</u>
<ul style="list-style-type: none"> <li>- Stark reduzierter Modellierungsaufwand</li> <li>- Modelle entsprechen stets dem Verhalten des tatsächlichen Systems</li> </ul>	<ul style="list-style-type: none"> <li>- Modelle weisen geringen Abstraktionsgrad auf</li> <li>- Analyse hinsichtlich NFEs bisher nicht möglich ohne Annahme bestimmter Ereignisse</li> <li>- Geringer Dokumentationscharakter</li> </ul>
<u>Möglichkeiten</u>	<u>Bedrohungen</u>
<ul style="list-style-type: none"> <li>- Automatisierte Erkennung von Anomalien im Verhalten</li> <li>- Hinzufügen von statischen Informationen zur besseren Interpretierbarkeit</li> </ul>	<ul style="list-style-type: none"> <li>- Fehlinterpretation der Modelle aufgrund des geringen Abstraktionsgrades</li> <li>- Unvollständig oder fehlerhaft beobachtetes Verhalten</li> </ul>

Aus diesen Überlegungen lässt sich folgende These formulieren:

**These**

Mit geringem manuellem Aufwand kann ohne spezifische Prozessinformationen und lediglich aus der Beobachtung von I/O-Signalen eine automatisierte Prozessanalyse zur Erkennung und Bewertung von Änderungen an Fertigungssystemen auf hoher Abstraktionsebene durchgeführt werden.

Anspruch der vorliegenden Arbeit ist es daher, Algorithmen zur automatischen Generierung von Verhaltensmodellen unter Verwendung von statischen a-priori-Informationen zu nutzen. Generierte Modelle sollen eine Analyse auf typisch relevanten Eigenschaften erlauben. Dabei sollen o.g. Anforderungen erfüllt werden. Die statischen Informationen sollten möglichst generell sein und die dynamischen Informationen lediglich aus I/O-Signalen bestehen. Ein Ansatz, der diesen Ansprüchen genügt, wird in den folgenden Kapiteln vorgestellt und evaluiert.

## 7 Allgemeiner Ansatz zur Wissensextraktion aus Ereignissequenzen

Ziel des hier vorgestellten Ansatzes ist es, aus einfachen Daten, die in Form von I/O-Signalverläufen vorliegen, Wissen in Form von Merkmalsausprägungen nicht-funktionaler Eigenschaften zu extrahieren. Dies geschieht über die Generierung von Modellen aus den Signalverläufen. Die ermittelten Merkmalsausprägungen können wiederum genutzt werden, um Anforderungen an die Systemqualität zu überprüfen. Diese Zusammenhänge sind in Abbildung 7-1 als UML-Klassendiagramm dargestellt.

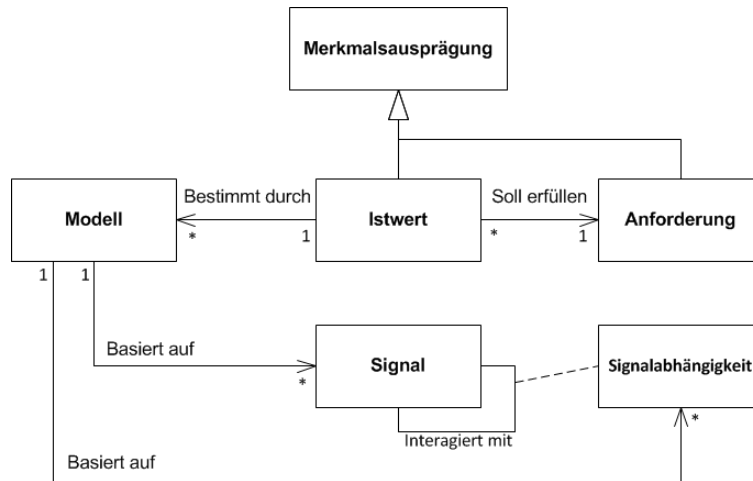


Abbildung 7-1: Vom Signal zur Merkmalsausprägung

Das Wissensmanagement [LIE99] beschreibt den Zusammenhang zwischen Daten und Wissen (Abbildung 7-2). Einfache Daten werden zu Informationen, wenn ihnen eine Bedeutung oder eine Struktur bzw. eine *Semantik* hinzugefügt wird. Information entspricht somit organisierten, strukturierten bzw. interpretierten Daten. Werden diese Informationen mittels Begründung, Abstraktion, Beziehung oder Anwendung in einen Zusammenhang gebracht, so wird dies als Wissen bezeichnet. Wissen liegt z.B. in Form eines Modells vor.

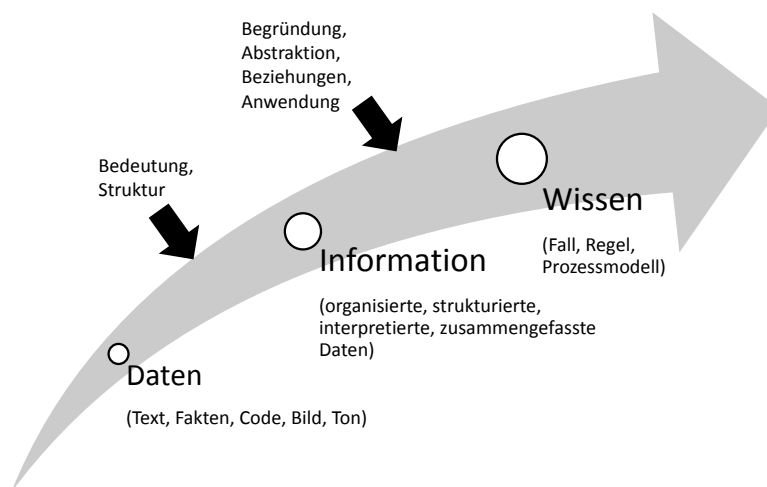


Abbildung 7-2: Wissenshierarchie des Wissensmanagements nach [LIE99]

Der folgende Abschnitt beschreibt eine Wissensarchitektur, die eine allgemeine Methode zur Wissensextraktion aus I/O-Daten entlang obiger Hierarchie beschreibt. Als Grundlage hierfür dient die Datenak-

quirierung, dessen Beschreibung und Rahmenbedingungen in Abschnitt 7.2 behandelt werden. Zur Hinzunahme von Semantik bedient sich der Ansatz einer Informationsmodellierung, die in Abschnitt 7.3 diskutiert wird. Dieses Kapitel schließt mit der Auswahl eines geeigneten Beschreibungsmittels zur Durchführung des Ansatzes ab (Abschnitt 7.4). Die für den Ansatz erarbeiteten Modelle und Algorithmen werden in den darauf folgenden Kapiteln (8-10) vorgestellt.

## 7.1 Wissensarchitektur

Entsprechend o.g. Wissenshierarchie ist der vorliegende Ansatz ausgerichtet. Die Methode wird dabei in Form einer Wissensarchitektur beschrieben (Abbildung 7-3). Die einzelnen Schritte der Wissensarchitektur von unten nach oben sind in den folgenden Absätzen beschrieben.

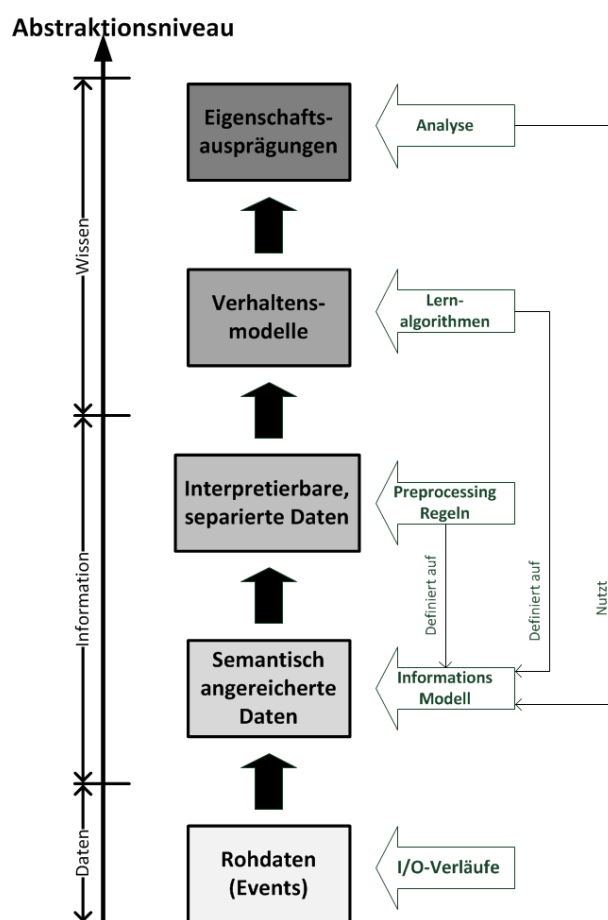


Abbildung 7-3: Wissensarchitektur des Ansatzes

Zunächst werden die Rohdaten aufgenommen (siehe Abschnitt 7.2). Die verwendeten Daten entsprechen im Rahmen dieser Arbeit gem. Anforderung 6-2 den im System aufkommenden I/O-Signalverläufen. Diese werden als einzige extern verfügbare Daten angenommen.

Die zur Interpretation der Daten notwendigen Semantiken werden aus einem Informationsmodell entnommen, mit denen die Daten angereichert werden. Das Informationsmodell beinhaltet die statischen Informationen und muss manuell erstellt werden. Um den manuellen Aufwand möglichst gering zu halten (Anforderung 6-3), beinhaltet das Informationsmodell lediglich minimal notwendige Semantiken, die zudem weitestgehend vordefiniert sind (siehe Abschnitt 7.3).

Basierend auf dem Informationsmodell kann zunächst eine Separierung der Daten hinsichtlich bestimmter Aspekte vorgenommen werden. So können mittels einfacher Regeln auf den Signalsemantiken für die Erstellung bestimmter Aspekt-Modelle lediglich diejenigen Signale berücksichtigt werden, die auch für den Aspekt relevant sind. Die Kapitel 8 und 9 beschreiben zwei Arten von Aspekt-Modellen, die sich für Fertigungssysteme eignen, und wie die Daten zur Generierung dieser separiert werden müssen.

Die separierten Signalverläufe werden in Lernalgorithmen zur Generierung der Verhaltensmodelle gespeist, um die vorhandenen Informationen in Beziehung zu bringen und Wissen zu gewinnen (Kapitel 8 und 9). Dabei greifen die Algorithmen zusätzlich auf das Informationsmodell zurück. Die Modelle werden zur Anomalieerkennung herangezogen. Die Anomalieerkennung dient der Erkennung von geänderterem Verhalten des Fertigungssystems und trägt daher der Erfüllung von Anforderung 6-1 bei. Grundsätzlich nutzt der hier beschriebene Ansatz Mechanismen, die aus der Fehlererkennung für DES als Teilgebiet der Fehlerdiagnose bekannt sind, vgl. bspw. [SLD11; SEA13; SEA14]. Diese wurden für die Aspekt-Modelle adaptiert und erweitert und sind in den entsprechenden Kapiteln erläutert. Allgemein betrachtet entspricht die Anomalieerkennung einem Abgleich beobachteten Verhaltens mit den gelernten Modellen. Wird Verhalten beobachtet, das durch keines der Modelle abgedeckt ist, so ist eine Anomalie erkannt. Dies kann zum einen ein beobachtetes Wort sein, das nicht in der Sprache der Modelle vorhanden ist, aber auch ein Zustandswechsel, der nicht in den Modellen vorhanden ist, oder eine Verletzung des gelernten zeitlichen Ablaufes. Eine erkannte Anomalie muss nicht zwingend aufgrund einer gewollten Änderung erkannt worden sein, sondern kann auch Resultat einer ungewollten Änderung (Seiteneffekt) oder eines Fehlers sein. Daher ist stets manuell zu bewerten, ob eine erkannte Anomalie Folge einer ungewollten Evolution oder eines Fehlers ist und behoben werden muss oder, ob es sich um eine gewollte Änderung handelt, die in die Modelle mit aufgenommen werden muss. Ein entsprechender Anpassungsprozess ist in [LHF+14B<sup>#</sup>] vorgeschlagen. Neben der Eignung zur Anomalieerkennung sind die Modelle so ausgerichtet, dass sie die Analyse hinsichtlich verschiedener NFEs zulassen (vgl. Anforderung 6-4). Die Analysealgorithmen nutzen zur Interpretation hinsichtlich der Qualitätseigenschaften sowohl die Modelle als auch die Signalverläufe und das Informationsmodell.

## 7.2 Datenakquirierung

Als Grundlage für die Verhaltensbeobachtungen dienen die I/Os der den Prozess steuernden SPSen. Die Inputs der SPSen entsprechen binären Sensorwerten z.B. von Näherungssensoren wie Lichtschranken und Endlagenschaltern. Die Outputs der SPSen entsprechen binären Schaltbefehlen, z.B. für Motoren. Wie durch Anforderung 6-2 eingegrenzt, ist kein Zugriff auf interne Variablen der Steuerungen oder des physikalischen Systems vorhanden (vgl. Abbildung 7-4).

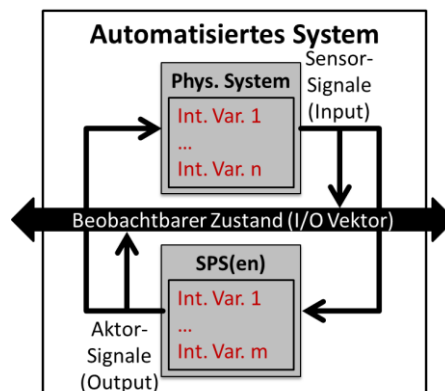


Abbildung 7-4: Beobachtbarer Zustand auf Basis der I/O-Signale

Um das Verhalten adäquat abbilden zu können, sind Zeitinformationen notwendig, vgl. Kapitel 4. Zudem beinhaltet eine Vielzahl der diskutierten Qualitätsmerkmale zeitliche Größen. Daher wird angenommen, dass neben der Abfolge von beobachteten Ereignissen auch eine Zeitstempelung dieser stattfindet, um das zeitliche Verhalten analysieren zu können. Eine mögliche Technologie zum Beobachten von Anlagendaten inklusive Zeitstempelung ist OPC [LIB10]. OPC bietet die Möglichkeit, Daten einer oder mehrerer Datenquellen (z.B. SPS) in einer Serverapplikation zu sammeln und Client-Applikationen (z.B. einem Datenlogger) zur Verfügung zu stellen. Viele Steuerungen haben heutzutage bereits einen OPC-Server integriert, was die Datenkommunikation erleichtert. Aufgrund dessen und, da es sich bei OPC um einen de facto-Standard handelt, der in der Industrie weit verbreitet ist (vgl. [LIB10]), wird hier davon ausgegangen, dass die Daten mittels OPC akquiriert werden und eine OPC-konforme Datenstruktur vorliegt. Eine mittels OPC generierte Ereignisliste enthält je Ereignis die folgenden Informationen:

1. Zeitstempel, der den Zeitpunkt des Ereignisses enthält
2. Name des Ereignis auslösenden Signals
3. Wert, den das Signal eingenommen hat (TRUE oder FALSE bzw. 0 oder 1)

Formal beschrieben bestehen die beobachteten I/Os aus  $n$  unterschiedlichen Signalen  $S_i$ ,  $i \in [0, n]$ . Jedes I/O-Signal entspricht einer binären Zustandsgröße und kann die Elementarzustände 0 und 1 annehmen. Sprachentheoretisch gesehen sind dies die *Zeichen* der Zustandsgrößen und die Menge  $\{0,1\}$  das *Alphabet* des Systems, vgl. [SEA13]. Der Wechsel des Zustandes einer Zustandsgröße zu einem bestimmten Zeitpunkt entspricht dem *Ereignis*. Jedes Signal kann also genau zwei Ereignisse hervorrufen, die als  $s_i^\uparrow(t)$  für steigende Flanken und  $s_i^\downarrow(t)$  für fallende Flanken des Signals zum Zeitpunkt  $t$  notiert sind. Der Vektor, der die Elementarzustände aller I/Os beinhaltet, beschreibt also die *Wörter* des Systems bzw. dessen Globalzustand und die Menge aller möglichen Wörter entsprechen der Sprache des Systems. Werden die I/Os partitioniert, werden nur Untermengen der Wörter betrachtet, da auch nur noch eine Untermenge der Signale bzw. Zustandsgrößen betrachtet wird. Somit wird auch nur eine Untermenge der Sprache betrachtet. Eine solche Untermenge wird im Folgenden als *Aspekt* bezeichnet.

Der OPC-Server kann Ereignisse nur innerhalb einer OPC-Zykluszeit  $T_{OPC}$  auflösen. Somit sind die Zeitstempel stets ganze Vielfache dieser Zykluszeit. Diese Vielfachen werden durch die Variable  $k$  ausgedrückt und ein Ereignis  $e_j(k)$  an der Stelle  $j$  der Ereignissequenz, das im Zyklus  $k$  beobachtet wurde, kann wie folgt beschrieben werden:

$$e_j(k) = \left( \begin{array}{c} s_i^\alpha \\ k \cdot T_{OPC} \end{array} \right), \alpha \in \{\uparrow, \downarrow\}, \forall i \in [0, n] \quad (7-1)$$

Während einer Beobachtungsdauer  $K \cdot T_{OPC}$  kann eine Ereignissequenz  $E_{obs}$  aufgenommen werden:

$$E_{obs} = \left( e_j(k) \right), 0 \leq k \leq K \quad (7-2)$$

Diese Sequenz liegt entsprechend der Zeitstempel in konsekutiver Reihenfolge vor. Sind die Zeitstempel mehrerer Ereignisse gleich, so ist deren Reihenfolge in der Ereignissequenz beliebig. Es wird hier angenommen, dass die Ereignissequenz der gesamten Sprache des Systems entspricht, d.h. dass das Verhalten vollständig und fehlerfrei beobachtet wurde.

### 7.3 Informationsmodellierung

Formal betrachtet entspricht das Informationsmodell einer Funktion, die jedem I/O-Signal einen Signaltypen und eine technische Ressource zuweist:

$$f : S \rightarrow ST \times R \quad (7-3)$$

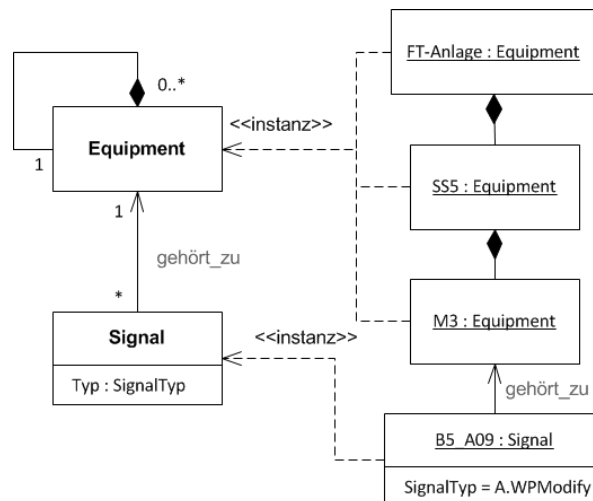
Mit:

**S**: Menge aller I/O-Signale

**ST**: Menge aller Signaltypen

**R**: Menge aller technischen Ressourcen

Die Struktur des Informationsmodells besteht aus einer Equipment-Hierarchie (Abbildung 7-5). Sie gibt an, wie sich die Anlage aus Subsystemen (z.B. Bereichen, Fertigungszellen o.ä.) und schlussendlich auf unterster Ebene der Hierarchie aus technischen Ressourcen (z.B. Maschinen, Transportbändern o.ä.) zusammensetzt. Die Signale werden stets denjenigen technischen Ressourcen zugeordnet, an denen sich die zugehörigen Sensoren und Aktoren befinden. Ein Signaltyp entspricht der Bedeutung des Signals. Das auf der rechten Seite von Abbildung 7-5 dargestellte Beispiel soll dies verdeutlichen. Hier ist ein Signal (*B5\_A09*) modelliert, das einen Motor ansteuert, der wiederum den Bohrer einer Werkzeugmaschine (*M3*) bewegt. Diese Maschine befindet sich in einer Teilanlage (*SS5*), die wiederum Teil der Gesamtanlage ist (*FT-Anlage*). Da es sich um einen Aktor handelt, ist dem Signal der Typ *A.WPModify* zugewiesen.



**Abbildung 7-5: Meta-Modell des Informationsmodells (links) und Beispiel (rechts) in UML-Notation**

Typischerweise umfasst eine informelle Spezifikation für Steuerungen Anzahl, Art und Wirkungsweise der verwendeten Aktoren und Sensoren [VDI 2221]. Diese Informationen befinden sich bspw. auch in Signallisten. Daher kann davon ausgegangen werden, dass die Signaltypen bekannt sind. Die im Informationsmodell verwendeten Signaltypen sollten so generisch sein, dass sie eine Vielzahl von Signalen abdecken und anlagenübergreifend verwendbar sind. Daher, und um den manuellen Aufwand zur Erstellung des Informationsmodells gering zu halten, wurden einige Signaltypen für Fertigungssysteme vordefiniert (Tabelle 7-1).

Zunächst wird unterschieden, ob es sich um ein Sensorsignal (SPS-Input) oder ein Aktorsignal (SPS-Output) handelt. Signaltypen für Aktoren besitzen den Präfix „A.“ und Signaltypen für Sensoren entsprechend „S.“. Die Aktorsignale werden unterschieden in Aktoren, die ein Werkstück greifen können, ein Werkstück bearbeiten können, einen Werkzeugwechsel veranlassen, eine technische Ressource bewegen oder allgemein ihren Zustand beeinflussen. Letzteres ist Signalen zuzuweisen, die keinem anderen Aktor-Signaltypen zugewiesen werden können. Die ersten beiden Aktor-Typen (*A.WPHold* und *A.WPModify*) indizieren die Interaktion mit einem Werkstück und können somit Rückschlüsse auf Transport und Bearbeitungsvorgänge geben. *A.Toolchange* lässt Rückschlüsse auf einen Werkzeugwechsel zu

und dient somit z.B. der Analyse hinsichtlich der Menge an durchführbaren Operationen. *A.Move* kann, in Kombination mit *A.WPHold*, zudem Informationen zu der Position eines Werkstückes geben, wenn eine bewegliche Ressource dieses transportiert.

**Tabelle 7-1: Vordefinierte Signaltypen und ihre Bedeutung**

<b>Signaltyp</b>	<b>Erläuterung</b>	<b>Beispiel</b>
<b><i>A.WPHold</i></b>	Aktor zum Greifen oder Loslassen eines Werkstückes	Pneumatischer Greifer ein/aus
<b><i>A.WPModify</i></b>	Aktor zur Bearbeitung eines Werkstückes	Bohrer ein/aus
<b><i>A.Toolchange</i></b>	Aktor zum Werkzeugwechsel	Drehen des Magazins einer Werkzeugmaschine
<b><i>A.Move</i></b>	Aktor zur Bewegung einer technischen Ressource	Drehen eines Drehtisches
<b><i>A.State</i></b>	Aktor, der allgemein den Zustand einer Ressource ändert	Werkstück einspannen
<b><i>S.State</i></b>	Sensor, der allgemein den Zustand einer technischen Ressource detektiert	Rückmeldung Motor an
<b><i>S.XPosition</i></b>	Sensor, der eine horizontale Position einer technischen Ressource detektiert	Endlagenschalter links/rechts
<b><i>S.YPosition</i></b>	Sensor, der eine vertikale Position einer technischen Ressource detektiert	Endlagenschalter oben/unten
<b><i>S.ZPosition</i></b>	Sensor, der eine horizontale Position einer technischen Ressource senkrecht zu <i>S.XPosition</i> detektiert	Endlagenschalter vorne/hinten
<b><i>S.WPDetect</i></b>	Sensor, der ein Werkstück detektiert	Lichtschranke
<b><i>S.WPIdentify</i></b>	Sensor, der einen Werkstücktypen identifiziert	Induktiver Sensor zur Erkennung metallischer Werkstücke
<b><i>S.WPHold</i></b>	Sensor zur Erkennung, dass eine technische Ressource ein Werkstück hält	Rückmeldung eines Greifers, dass ein Werkstück gegriffen ist
<b><i>Auxiliary</i></b>	Signale, die nicht dem technischen Fertigungsprozess zugewiesen werden können	Aktorsignal zum Einschalten einer Signalleuchte

Die Sensorsignale werden unterschieden in Sensoren, die allgemein den Zustand einer technischen Ressource wiedergeben (ähnlich wie *A.State*), die Position einer technischen Ressource wiedergeben (ähnlich wie *A.Move*), anhand ihrer Materialeigenschaften Werkstücke identifizieren, ein Werkstück detektieren, oder das Halten eines Werkstückes durch eine technische Ressource indizieren (ähnlich wie *A.WPHold*). Insbesondere *S.WPIdentify* und *S.WPDetect* dienen der Ermittlung des Materialflusses und innerhalb dessen der Unterscheidung in Werkstücktypen (vgl. Kapitel 9). Signale, die keine Aussage über das fertigungsrelevante Verhalten des Systems zulassen, sind dem Signaltypen *Auxiliary* zuzuweisen. Alternativ kann auf das Beobachten von diesen Signalen verzichtet werden.

## 7.4 Auswahl des Beschreibungsmittels

Da kein Beschreibungsmittel in der Lage ist, sämtliche Aspekte eines Systems zu beschreiben [FAY09], ist stets eine lösungsorientierte Auswahl zu treffen. Die in Kapitel 4 vorgestellten Beschreibungsmittel sollen daher im Folgenden auf Ihre Eignung für den hier vorzustellenden Ansatz bewertet werden. Als Grundlage hierfür dient die [VDI/VDE 3681], welche Beschreibungsmittel anhand verschiedener Eigenschaften und Einsatzzwecke evaluiert. Um auch Beschreibungsmittel bewerten zu können, die nicht in der [VDI/VDE 3681] behandelt werden, wird weitere Literatur hinzugezogen. Göring [GÖR13] hat sich mit der Bewertung der PSL entsprechend der Bewertungskriterien nach [VDI/VDE 3681] beschäftigt. Die PSL wird hier aufgrund ihrer prinzipiellen Eignung ebenfalls betrachtet. Die Bewertung von Zustands- und Aktivitätsdiagrammen basiert auf den Diskussionen in Abschnitt 4.1.3. Zudem werden die Diskussionen zu UML und SysML bzw. State Charts von Göring [GÖR13] und Schnieder [SCH99] berücksichtigt.

Die in der [VDI/VDE 3681] beschriebenen, hier relevanten Kriterien und Unterscheidungsmerkmale sind:

1. Formale Basis: Unterscheidet im zugrundeliegenden Formalismus des Beschreibungsmittels
  - Formal: Eindeutige Syntax und Semantik, mathematisch-formale Basis
  - Semi-formal: Eindeutige Syntax und Semantik, keine mathematisch-formale Basis
2. Verhaltensbeschreibung: Art der Darstellung des Verhaltens
  - Deterministisch: Eindeutig vorhersagbares Verhalten
  - Nicht-deterministisch: Nicht-eindeutig vorhersagbares Verhalten
  - Statisch: Erzeugt stets bei gleicher Eingangsbelegung das gleiche Ausgangswort
  - Dynamisch: Ausgangswort abhängig vom Eingang und vorherigem Zustand
3. Zeitdarstellung: Untersucht, wie Zeit in die Modellierung einfließt
  - Ereignisgetrieben diskret: Zeit wird als Dauer zwischen Ereignissen dargestellt
  - Zeitdiskret: Zeit wird als Anzahl von Ereignissen oder eines Taktes angegeben
  - Zeitkontinuierlich: Zeit geht als kontinuierliche Variable in das Modell ein
4. Synchronisation: Verkoppeln von Teil-Modellen
  - Synchron: Teilprozesse werden über Taktsignal synchronisiert, laufen gleichzeitig ab
  - Asynchron: Teilmodelle laufen mit jeweils eigenem Takt
  - Nebenläufig: Prozesse laufen kausal unabhängig voneinander ab
5. Darstellung: Wie das Beschreibungsmittel dokumentiert werden kann
  - Textuell
  - Mathematisch-symbolisch
  - Grafisch

Eine formale Basis wird zum einen benötigt, um Lernalgorithmen zu definieren, aber auch um automatisierte Analysen auf den erlernten Modellen durchzuführen. Semi-formale Modelle eignen sich hierzu nur, wenn ihre Semantik weiter verfeinert oder formalisiert wird (vgl. Abschnitt 5.1). Somit ist semi-Formalismus kein Ausschlusskriterium, jedoch sind formale Beschreibungsmittel vorzuziehen. Ein für diesen Ansatz geeignetes Beschreibungsmittel muss in der Lage sein, dynamisches Verhalten darzustellen. Es sollte sich zur eindeutigen Dokumentation des Systemverhaltens eignen. Somit sollten mithilfe des Beschreibungsmittels deterministische Modelle erstellt werden können. Nichtsdestotrotz können Zusammenhänge vorherrschen, die statistisch verteilt sind, oder, die durch einen Lernalgorithmus nicht erfasst werden. Nicht erfasste Zusammenhänge können anhand stochastischer Beschreibungen angenähert werden. Daher sollte außerdem die Möglichkeit bestehen, nicht-deterministisches Verhalten darzustellen. Idealerweise sollte die Zeit als kontinuierliche Variable einfließen. Dies hat den Grund, dass viele der diskutierten NFEs zeitabhängig sind bzw. Zeitvariablen zur Beschreibung benötigen. Eine zeitdiskrete

Darstellung ist ebenfalls möglich, würde jedoch die Zeitauflösung und somit die Genauigkeit der Analyseergebnisse reduzieren. Wie bereits beschrieben, soll u.a. der Materialfluss innerhalb des Fertigungssystems beschrieben werden. Dieser ist typischerweise hoch nebenläufig und muss entsprechend dargestellt werden können. Das Beschreibungsmittel sollte die Möglichkeit zur grafischen Darstellung bieten, um als Dokumentation verwendet werden zu können, die für den Menschen leicht interpretierbar ist.

Das Ergebnis der Bewertung ist in Tabelle 7-2 dargestellt. Die Bewertung der Beschreibungsmittel findet anhand eines Punktesystems statt. Besitzt ein Beschreibungsmittel eine für diesen Ansatz ideale Eigenschaft (Spalten jeweils in dunkelgrau dargestellt), so wird das Beschreibungsmittel für dieses Kriterium mit zwei Punkten bewertet. Besitzt es eine nicht ideale, aber dennoch für den Ansatz brauchbare Eigenschaft (Spalten jeweils in hellem grau dargestellt), so wird es mit einem Punkt bewertet. Pro Kriterium kann ein Beschreibungsmittel jedoch mit maximal zwei Punkten bewertet werden. Dies verhindert, dass ein Beschreibungsmittel, das eine ideale Eigenschaft und zusätzlich eine brauchbare Eigenschaft desselben Kriteriums enthält, mit mehr als zwei Punkten bewertet wird. Somit kann ein Beschreibungsmittel mit maximal zehn Punkten bewertet werden. Es ist allerdings zwingend erforderlich, dass ein brauchbares Beschreibungsmittel in jedem Kriterium mindestens einen Punkt aufweist und zudem in der Lage ist, dynamisches Verhalten abzubilden. Das Bewertungsergebnis als Summe aller Punkte ist in der letzten Spalte von Tabelle 7-2 angegeben. Eine durchgestrichene Zahl bedeutet, dass das Beschreibungsmittel aufgrund eines 0-Punkte Kriteriums ausgeschlossen wurde.

**Tabelle 7-2: Bewertung der diskutierten Beschreibungsmittel**

Kriterium \ Beschreibungsmittel	Formale Basis		Verhaltensbeschreibung			Zeitarstellung			Synchronisation			Darstellung			Bewertungsergebnis	
	Formal	Semi-Formal	Deterministisch	Nicht-Deterministisch	Statisch	Dynamisch	Ereignisgetrieben diskret	Zeitdiskret	Zeitkontinuierlich	Synchron	Asynchron	Nebenläufig	Textuell	Mathematisch-Symbolisch		Grafisch
Zustandsautomaten	x		x	x		x	x	x	x	x			x	x	x	<del>8</del>
Petri-Netze	x	x	x	x		x	x	x	x			x	x <sup>1</sup>		x	10
Aktivitätsdiagramm		x		x		x	x	x	x		x	x			x	9
Zustandsdiagramm		x		x		x	x	x		x	x	x			x	8
Algebraische Modelle	x		x	x	x	x	x	x		x	x			x		<del>4</del>
PSL	x		x		x		x	x			x	x	x			<del>6</del>

<sup>1</sup>: Über Austauschformat PNML [ISO/IEC 15909] (hier ergänzt)

Ideale Eigenschaft (2 Punkte)
Brauchbare Eigenschaft (1 Punkte)

Das Bewertungsergebnis zeigt, dass Petri-Netze für den hier vorzustellenden Ansatz am geeignetsten scheinen. Sie besitzen alle ideal-Eigenschaften und darüber hinaus einige weitere brauchbare Eigenschaften.

## 8 Maschinenzustandsmodelle: Definition, Lernalgorithmus und Anomalieerkennung

Dieser Abschnitt beschreibt den ersten Typ Aspekt-Modelle, die für den vorgestellten Ansatz verwendet werden. Sie bilden für jede technische Ressource deren vollständiges I/O-Verhalten ab. Maschinenzustandsmodelle erlauben damit, das Verhalten einzelner Ressourcen unabhängig der anderen analysieren zu können. Zum anderen enthalten sie möglichst alle kausalen Zusammenhänge innerhalb einer technischen Ressource und dienen daher besonders gut zur Erkennung geänderten Verhaltens einzelner Ressourcen. Entsprechend beinhalten die Modelle sämtliche I/O-Ereignisse, die der Ressource zugewiesen werden können. Da die Modelle das Verhalten in Form von Maschinenzustandsübergängen beschreiben und auf Petri-Netzen basieren, werden sie im Folgenden als *Machine State Petri Nets (MSPN)* bezeichnet. Der folgende Abschnitt gibt zunächst eine formale Beschreibung der MSPN. Darauf aufbauend wird der Lernalgorithmus zur MSPN-Generierung beschrieben. Das Kapitel schließt mit einer Diskussion zur Anomalieerkennung mittels MSPN ab.

### 8.1 Definition

Ein MSPN ist ein Petri-Netz,  $\mathcal{PN} = \langle \mathcal{S}, \mathcal{T}, \mathcal{F}, m \rangle$ , mit einer Menge an Stellen  $\mathcal{S}$ , einer Menge an Transitionen  $\mathcal{T}$  und einer Menge an Flussrelationen  $\mathcal{F} \subseteq (\mathcal{S} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S})$  sowie einer Menge Marken  $m$ , die den Stellen zugeordnet sind. Die Markierung einer Stelle ist binär. Der *Vorbereich*  $\bullet\mathcal{T}_t$  einer Transition  $\mathcal{T}_t$  entspricht denjenigen Stellen, von denen aus eine Kante auf die Transition zeigt:

$$\bullet\mathcal{T}_t = \{ \mathcal{S} \in \mathcal{S} : (\mathcal{S}, \mathcal{T}_t) \in \mathcal{F} \} \quad (8-1)$$

Die Stellen des Vorbereiches einer Transition werden im Folgenden als *Vorstellen* der Transition bezeichnet. Analog dazu entsprechen diejenigen Stellen, auf die eine von  $\mathcal{T}_t$  ausgehende Kante zeigt, dem *Nachbereich*  $\mathcal{T}_t\bullet$  der Transition:

$$\mathcal{T}_t\bullet = \{ \mathcal{S} \in \mathcal{S} : (\mathcal{T}_t, \mathcal{S}) \in \mathcal{F} \} \quad (8-2)$$

Die Stellen des Nachbereiches einer Transition werden im Folgenden als *Nachstellen* der Transition bezeichnet. Der Term  $|\bullet\mathcal{T}_t(\mathcal{S})|$  beschreibt die Kardinalität des Vorbereiches einer Transition, also die Anzahl an Vorstellen der Transition  $\mathcal{T}_t$  aus der Menge  $\mathcal{S}$ . Analog dazu beschreibt der Term  $|\mathcal{T}_t\bullet(\mathcal{S})|$  die Kardinalität des Nachbereiches, also die Anzahl an Nachstellen der Transition  $\mathcal{T}_t$  aus der Menge  $\mathcal{S}$ .

Eine Transition wird als *aktiviert* bezeichnet, wenn alle Stellen ihres Vorbereiches mit einer Marke besetzt sind und alle Stellen ihres Nachbereiches nicht mit einer Marke besetzt sind:

$$\mathcal{T}_t \text{ aktiviert} \Leftrightarrow (\forall \mathcal{S} \in \bullet\mathcal{T}_t : m(\mathcal{S}) = 1) \wedge (\forall \mathcal{S} \in \mathcal{T}_t\bullet : m(\mathcal{S}) = 0) \quad (8-3)$$

Der Vorgang alle Marken aus dem Vorbereich einer Transition zu entfernen und alle Stellen des Nachbereiches mit Marken zu besetzen, wird als *feuern* bezeichnet. Somit ergibt sich bei gegebener Markierung  $m_k$  nach dem Schaltvorgang einer aktivierten Transition  $\mathcal{T}_t$  die folgende Markierung:

$$m_{k+1}(\mathcal{S}) = \begin{cases} m_k(\mathcal{S}), & \text{für } (\mathcal{S} \notin \bullet\mathcal{T}_t) \wedge (\mathcal{S} \notin \mathcal{T}_t\bullet) \\ 1, & \text{für } \mathcal{S} \in \mathcal{T}_t\bullet \\ 0, & \text{für } \mathcal{S} \in \bullet\mathcal{T}_t \end{cases} \quad (8-4)$$

Jede Transition ist mit einem (kombinierten) Ereignis  $\tilde{e}(\mathcal{T}_t)$  annotiert. Dieses Ereignis besteht aus einem oder mehreren Ereignissen der Ereignissequenz  $E_{obs}$  (siehe Abschnitt 7.2) und beschreibt somit eine Menge an Zustandswechseln von I/O-Signalen. Zudem ist jede Transition mit einem 5-Tupel  $\langle d_{min}, d_{max}, \mu, \sigma, n \rangle$  annotiert, welches das Verhalten des MSPN weiter spezifiziert. Die Elemente des 5-Tuples sind wie folgt definiert:

- $d_{min}$ : Minimale Aktivierungsdauer der Transition
- $d_{max}$ : Maximale Aktivierungsdauer der Transition
- $\mu$ : Mittlere Aktivierungsdauer der Transition
- $\sigma$ : Standardabweichung der Aktivierungsdauer der Transition
- $n$ : Anzahl an Feuervorgängen (s.u.) während der Beobachtung

Eine Transition  $\mathcal{T}_t$  eines MSPN feuert, wenn die folgenden Bedingungen erfüllt sind:

- (1) Die Transition ist aktiviert
- (2) Die Aktivierungsdauer liegt zwischen  $d_{min}$  und  $d_{max}$
- (3) Das annotierte Ereignis  $\tilde{e}(\mathcal{T}_t)$  der Transition tritt auf. Besteht  $\tilde{e}(\mathcal{T}_t)$  aus mehreren Ereignissen, müssen diese innerhalb einer vordefinierten Schwelldauer  $T_{thresh}$  auftreten (vgl. Abschnitt 8.2)

Die Menge an Stellen setzt sich aus zwei Untermengen zusammen:

$$\mathcal{S} = \mathcal{S}_{sig} \cup \mathcal{S}_{mach} \quad (8-5)$$

$\mathcal{S}_{mach}$  wird hierbei als Menge an Stellen mit Bezug zum Maschinenverhalten verstanden und  $\mathcal{S}_{sig}$  als Menge an Stellen mit Bezug zum Signalverhalten. Die beiden Mengen sind disjunkt und es gilt:

$$\mathcal{S}_{sig} \cap \mathcal{S}_{mach} = \{\emptyset\} \quad (8-6)$$

Die Menge an Flussrelationen ist in gleicher Weise aus Untermengen zusammengesetzt:

$$\mathcal{F} = \mathcal{F}_{sig} \cup \mathcal{F}_{mach} \quad (8-7)$$

$$\mathcal{F}_{sig} \cap \mathcal{F}_{mach} = \{\emptyset\} \quad (8-8)$$

Die Flussrelationen  $\mathcal{F}_{sig} \subseteq (\mathcal{S}_{sig} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S}_{sig})$  entsprechen Kanten zwischen Transitionen und auf Signalverhalten bezogenen Stellen. Ähnlich dazu, beschreiben die Flussrelationen  $\mathcal{F}_{mach} \subseteq (\mathcal{S}_{mach} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S}_{mach})$  Kanten zwischen Transitionen und auf Maschinenverhalten bezogenen Stellen.

Es sei erwähnt, dass beide Teile des MSPN (d.h. der Teil mit Bezug auf Maschinenverhalten sowie der Teil mit Bezug auf Signalverhalten) stets die gesamte Menge an Transitionen  $\mathcal{T}$  umfassen. Der Teil eines MSPN mit Bezug auf Maschinenverhalten  $\langle \mathcal{S}_{mach}, \mathcal{T}, \mathcal{F}_{mach} \rangle$  beschreibt die Interaktion der I/O-Signale. Ergänzend beschreibt der Teil mit Bezug auf Signalverhalten  $\langle \mathcal{S}_{sig}, \mathcal{T}, \mathcal{F}_{sig} \rangle$  die Eigenschaft der Signale, stets nur zwischen ihren beiden Elementarzuständen (0 und 1) wechseln zu können. Hierzu repräsentiert jede Stelle aus  $\mathcal{S}_{sig}$  einen Elementarzustand eines Signals. Ist die Stelle markiert, so bedeutet dies, dass sich das zugehörige Signal in dem entsprechenden Zustand befindet. Eine Transition, an der ein Ereignis in Form eines Zustandswechsels des Signals annotiert ist, muss einen entsprechenden Markenfluss zwischen den beiden Zuständen hervorrufen. Somit gewährleistet der auf Signalverhalten bezogene Teil, dass auf jede steigende Flanke eines Signals  $s_i^\uparrow$  stets eine fallende Flanke  $s_i^\downarrow$  folgen muss, bevor

die nächste steigende Flanke auftreten kann. Dies reduziert die Permissivität des gelernten Modells. Zugleich beschreibt die Markierung des auf Signalverhalten bezogenen Teils stets ein Wort der Sprache der Ressource.

Für die Struktur eines MSPN gelten die folgenden strukturellen Einschränkungen:

$$|\bullet\mathcal{T}_t(\mathcal{S}_{mach})| = 1, \forall t \quad (8-9)$$

$$|\mathcal{T}_t\bullet(\mathcal{S}_{mach})| = 1, \forall t \quad (8-10)$$

$$|\bullet\mathcal{T}_t(\mathcal{S}_{sig})| \geq 1, \forall t \quad (8-11)$$

$$|\mathcal{T}_t\bullet(\mathcal{S}_{sig})| \geq 1, \forall t \quad (8-12)$$

$$|\mathcal{T}_t\bullet(\mathcal{S}_{sig})| = |\bullet\mathcal{T}_t(\mathcal{S}_{sig})|, \forall t \quad (8-13)$$

Dies bedeutet, dass jede Transition exakt eine Vorstelle (8-9) und exakt eine Nachstelle (8-10) aus der Menge Stellen mit Bezug zum Maschinenverhalten besitzt sowie mindestens eine Vorstelle (8-11) und mindestens eine Nachstelle (8-12) aus der Menge an Stellen mit Bezug zum Signalverhalten. Zudem besitzt eine Transition stets dieselbe Anzahl an Stellen mit Bezug zum Signalverhalten im Vor- wie im Nachbereich (8-13). Aus diesen Einschränkungen ergibt sich, dass die Anzahl an markierten Stellen in  $\mathcal{S}_{mach}$  ebenso wie die Anzahl an markierten Stellen in  $\mathcal{S}_{sig}$  konstant sind. Entsprechend ist die Anzahl an markierten Stellen im gesamten MSPN konstant.

Neben einer grafischen Darstellung werden Petri-Netz Strukturen häufig in Form einer *Inzidenzmatrix*  $\mathcal{C}$  beschrieben. Sie ergibt sich aus einer *Eingangsinzidenzmatrix*  $\mathcal{C}^+$  und einer *Ausgangsinzidenzmatrix*  $\mathcal{C}^-$ . Die Spalten der Matrizen verweisen dabei auf die Transitionen des Petri-Netzes und die Zeilen auf die Stellen. Die Eingangsinzidenzmatrix hat an jedem Matricelement  $(i,j)$  einen 1-Eintrag, wenn die Stelle  $\mathcal{S}_i$  zum Nachbereich der Transition  $\mathcal{T}_j$  gehört:

$$\mathcal{C}^+ = (c_{i,j}), \quad c_{i,j} = \begin{cases} 1, & \text{für } \mathcal{S}_i \in \mathcal{T}_j\bullet \\ 0, & \text{sonst} \end{cases} \quad (8-14)$$

Analog dazu hat die Ausgangsinzidenzmatrix dort einen 1-Eintrag, wo die Stellen zum Vorbereich der Transitionen gehören:

$$\mathcal{C}^- = (c_{i,j}), \quad c_{i,j} = \begin{cases} 1, & \text{für } \mathcal{S}_i \in \bullet\mathcal{T}_j \\ 0, & \text{sonst} \end{cases} \quad (8-15)$$

Die Inzidenzmatrix ergibt sich wie folgt:

$$\mathcal{C} = \mathcal{C}^+ - \mathcal{C}^- \quad (8-16)$$

Entsprechend hat eine Inzidenzmatrix immer dann einen Eintrag von -1, wenn die Stelle, die durch die Zeile repräsentiert wird, Teil des Vorbereiches derjenigen Transition ist, die durch die Spalte repräsentiert wird. Analog ist der Eintrag 1, wenn die Stelle Teil des Nachbereiches der Transition ist.

## 8.2 Preprocessing der Ereignissequenz

Bevor die MSPNs generiert werden können, ist zunächst ein Preprocessing der zugrundeliegenden Ereignissequenz notwendig. Für die Maschinenzustandsmodelle wird die I/O-Sequenz entsprechend des Aspektes der Zugehörigkeit der Signale zu den technischen Ressourcen partitioniert. Dies geschieht automatisch mithilfe des Informationsmodells. Somit wird für jede Ressource  $R_r$  eine Ereignissequenz  $E_{obs}^r$  generiert, die lediglich Ereignisse  $e_j^r(k)$  derjenigen Signale enthält, die der zugehörigen technischen Ressource zugewiesen wurden und nicht den Signaltypen *Auxiliary* besitzen:

$$e_j^r(k) = \left\{ \left( \begin{array}{c} S_i^\alpha \\ k \cdot T_{OPC} \end{array} \right) : R(S_i) = R_r \wedge ST(S_i) \neq \text{auxiliary} \right\} \quad (8-17)$$

Mit:

$R(S_i)$ : Technische Ressource, der im Informationsmodell Signal  $S_i$  zugewiesen wurde

$ST(S_i)$ : Im Informationsmodell Signal  $i$  zugewiesener Signaltyp

In einem zweiten Preprocessing-Schritt werden Ereignisse der Ereignissequenz zusammengeführt, wenn sie innerhalb einer definierten Zeitschwelle  $T_{thresh}$  auftreten. Hiermit werden zwei Punkte in der Anwendung des Algorithmus adressiert. Zum einen wird berücksichtigt, dass Ereignisse mit gleichem Zeitstempel nacheinander in beliebiger Reihenfolge in der Ereignissequenz auftreten können (siehe Abschnitt 7.2). Dies führt bei der Modellgenerierung zu einer nicht aus Verhalten resultierenden und somit unnötigen Komplexität des resultierenden Modells. Daher werden Ereignisse mit gleichem Zeitstempel zu kombinierten Ereignissen zusammengeführt. Zum anderen ist mit einer technisch bedingten Streuung bzw. Unschärfe der Zeitstempel zu rechnen. Diese Unschärfe kann mehrere Gründe haben:

- (1) Es können, aufgrund unterschiedlicher Signalübertragungswege und -Technologien, unterschiedliche Latenzzeiten in der Übertragung von Signalinformationen vorherrschen
- (2) Die Zykluszeit des OPC-Servers kann sich z.T. stark von der oder den Zykluszeit/en der SPS/en unterscheiden, was zu Ungenauigkeiten in der Zeitstempelung führt. Dieser Effekt verstärkt sich zusätzlich, wenn die SPS-Zykluszeiten nicht konstant sind
- (3) Weitere Streuungen von Latenzzeiten können aufgrund von Rechen- und Kommunikationszeiten oder weiteren physikalischen Effekten im Automatisierungssystem auftreten

Um diese Unschärfe herauszufiltern, wird eine Zeitschwelle als positive ganze Vielfache der Zyklusdauer des OPC-Servers  $T_{OPC}$  definiert:

$$T_{thresh} = u \cdot T_{OPC}, \quad u \in \mathbb{N}^+ \quad (8-18)$$

Alle Ereignisse, deren Zeitstempel eine Differenz kleiner dieser Schwelle aufweisen, werden zu einem kombinierten Ereignis zusammengefasst:

$$\tilde{e}_j^r(k) = \{e_j^r(k), \dots, e_{j+m}^r(k+v) : v < u\}, \quad v \in \mathbb{N}^+ \quad (8-19)$$

Hieraus wird eine neue Ereignissequenz generiert:

$$E^r = (\tilde{e}_j^r(k)) \quad (8-20)$$

Vereinfachend wird der Index  $j$  entsprechend der Position des Ereignisses in  $E^r$  neu definiert. Das beschriebene Zusammenführen der Ereignisse wirkt wie ein Tiefpass auf die Ereignissequenz, da es Dynamiken mit Zeiten kleiner oder gleich  $T_{thresh}$  ausblendet. Dies führt dazu, dass die resultierende Ereignis-

sequenz diese Dynamiken nicht mehr enthält. Da typischerweise Latenzzeiten sehr viel kleiner sind als Dynamiken von Fertigungssystemen, kann der Wegfall schneller Dynamiken hier jedoch akzeptiert werden.

### 8.3 Generieren des Petri-Netzes

Ziel ist es, für jede technische Ressource ein MSPN zu generieren, das dieselbe Sprache enthält, wie die Ressource im Beobachtungszeitraum. Mit anderen Worten heißt dies, dass die Ereignissequenzen, die durch das Verhalten (Feuersequenzen) eines MSPN beschrieben werden, Subsequenzen der zugehörigen Ereignisliste  $E^r$  entsprechen. Der hier präsentierte Algorithmus wird entsprechend für jede technische Ressource bzw. jede zugehörige Ereignissequenz durchgeführt. Die Generierung des MSPN besteht dabei aus drei Teilen. Die ersten beiden Teile erzeugen die Struktur des MSPN. Zunächst wird der auf Maschinenverhalten bezogene Teil  $\langle \mathcal{S}_{mach}, \mathcal{T}, \mathcal{F}_{mach} \rangle$  aus der Ereignissequenz  $E^r$  generiert. Im Anschluss wird die Permissivität des Modells reduziert, d.h. die Sprache des Systems wird weiter eingeschränkt, indem der auf Signalverhalten bezogene Teil hinzugefügt wird. In einem letzten Schritt werden die Annotationen  $\langle d_{min}, d_{max}, \mu, \sigma, n \rangle$  aus der Ereignissequenz abgeleitet.

#### 8.3.1 Generieren des auf Maschinenverhalten bezogenen Teils

Die Strukturgenerierung des auf Maschinenverhalten bezogenen Teils basiert auf der Kausalitätsspezifikation von Lefebvre und Leclercq [LELE11]. Es sei erwähnt, dass hier zwar eine andere Notation verwendet wird, jedoch derselbe Algorithmus beschrieben wird. Der Algorithmus besteht aus 4 Stufen:

##### 1. Ermittlung einer Ereignispropagationsmatrix

Zunächst wird aus der Ereignissequenz eine Ereignispropagationsmatrix  $B(E^r)$  ermittelt. Sie enthält die Information, ob ein Ereignis in der Sequenz ein direkter Nachfolger eines anderen Ereignisses ist. Hierzu wird sie zunächst als Nullmatrix der Dimension  $\nu \times \nu$  initialisiert. Die natürliche Zahl  $\nu$  ist hierbei als die Anzahl unterschiedlicher Ereignisse in  $E^r$  definiert. Jede Zeile (respektive jede Spalte) verweist dabei auf eines dieser Ereignisse. Jedes Element der Matrix  $b_{uv} \in B(E^r)$  wird mit einer 1 beschrieben, wenn das Ereignis, das von Spalte  $\nu$  referenziert wird, mindestens einmal direkter Nachfolger desjenigen Ereignisses ist, das von Zeile  $u$  referenziert wird:

$$b_{uv} = \begin{cases} 1, & \text{wenn Ereignis } u \text{ in } E^r \text{ mind. einmal direkt auf Ereignis } v \text{ folgt} \\ 0, & \text{sonst} \end{cases} \quad (8-21)$$

##### 2. Berechnen einer Inzidenzmatrix $\mathcal{C}$

In diesem Schritt wird ein Petri-Netz generiert, dessen Transitionen mit denselben Ereignissen annotiert sind, wie von  $B(E^r)$  referenziert. Die Teil-Ereignissequenzen in  $E^r$  sollen von dem Ziel-Petri-Netz anhand möglicher Reihenfolgen von Feuervorgängen von Transitionen mit entsprechender Annotation repräsentiert werden. Das heißt, folgen zwei Ereignisse  $\tilde{e}_n^r(k)$  und  $\tilde{e}_{n+1}^r(k)$  in  $E^r$  direkt aufeinander, so muss die Nachstelle einer Transition, die mit  $\tilde{e}_n^r(k)$  annotiert ist, auch Vorstelle einer Transition sein, die mit  $\tilde{e}_{n+1}^r(k)$  annotiert ist, so dass die beiden Transitionen in direkter Nachfolge feuern können. Der Algorithmus weist zunächst jedem unterschiedlichen Ereignis in  $E^r$  genau eine Transition und jeder Transition genau eine Nachstelle zu. Dies lässt sich mithilfe einer Einheitsmatrix der Größe  $\nu \times \nu$  als Inzidenzmatrix beschreiben. Die Transitionen des durch diese Inzidenzmatrix beschriebenen Petri-Netzes (d.h. die Spalten der Inzidenzmatrix) werden nun mit den Ereignissen annotiert, auf die die Zeilen/Spalten der Ereignispropagationsmatrix verweisen. Da die Inzidenzmatrix zunächst als Einheits-

matrix beschrieben ist, hat die Transition, die durch Spalte  $n$  der Inzidenzmatrix repräsentiert wird (annotiert mit Ereignis  $\tilde{e}_n^r(k)$ ), diejenige Stelle als Nachstelle, die durch Zeile  $n$  der Inzidenzmatrix repräsentiert wird. Soll diese Stelle ebenfalls als Vorstelle der Transition  $n+1$  dienen, so muss die Inzidenzmatrix an der Stelle  $(n, n + 1)$  einen Eintrag von  $-1$  vorweisen. Folgt Ereignis  $\tilde{e}_{n+1}^r(k)$  direkt auf  $\tilde{e}_n^r(k)$ , so hat die Ereignispropagationsmatrix  $B(E^r)$  an der Stelle  $(n + 1, n)$  einen Eintrag von  $1$ . Folglich lässt sich eine Inzidenzmatrix, die alle Feuersequenzen generieren kann, die in  $E^r$  enthalten sind, mittels folgender Gleichung berechnen.

$$C = I_\nu - B(E^r)^T \tag{8-22}$$

Mit:

$I_\nu$ : Einheitsmatrix der Größe  $\nu \times \nu$

Hierbei sind die Transitionen, respektive die Spalten von  $C$ , entsprechend der Zuweisung der Spalten von  $B(E^r)$  zu Ereignissen zu annotieren.

Folgendes Beispiel soll dies verdeutlichen:

Es sei eine Ereignissequenz mit genau zwei Ereignissen  $\tilde{e}_1^r$  und  $\tilde{e}_2^r$  in dieser Reihenfolge gegeben. Entsprechend ergibt sich folgende Ereignispropagationsmatrix:

$$B(E^r) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \tag{8-23}$$

Ein Petri-Netz, das die gegebene Ereignissequenz in Form von Feuervorgängen annotierter Transitionen repräsentiert, benötigt zwei Transitionen, eine mit  $\tilde{e}_1^r$  annotiert und eine mit  $\tilde{e}_2^r$  annotiert. Hierbei muss die Nachstelle der mit  $\tilde{e}_1^r$  annotierten Transition ebenso Vorstelle der mit  $\tilde{e}_2^r$  annotierten Transition sein. Dieses Ziel-Petri-Netz ist in Abbildung 8-1 dargestellt, wobei  $T1$  mit Ereignis  $\tilde{e}_1^r$  zu annotieren ist und  $T2$  mit Ereignis  $\tilde{e}_2^r$ .

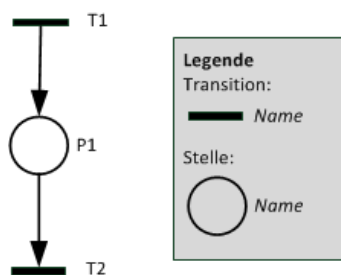


Abbildung 8-1: Ziel-Petri-Netz-Struktur des Beispiels

Ausgehend von einer Inzidenzmatrix in Form einer Einheitsmatrix der Größe  $2 \times 2$ , ist  $P1$  Nachstelle von  $T1$  und  $P2$  Nachstelle von  $T2$ <sup>29</sup>. Damit Stelle  $P1$  ebenfalls Vorstelle von Transition  $T2$  ist, muss eine  $-1$  an der Stelle  $(1, 2)$  eingefügt werden. Dies ergibt sich exakt, wenn die Transponierte von  $B(E^r)$  von der Einheitsmatrix abgezogen wird. Für eine tiefere Diskussion hierzu sei auf [LeLe11] verwiesen.

### 3. Stellen-Reduktion und Transitionen-Duplikation

In diesem Schritt wird die Struktur des bisherigen Petri-Netzes verändert. Zum einen wird die Komplexität reduziert, indem Stellen zusammengeführt werden. Zum anderen werden Transitionen dupliziert, um einen Zustandsgraphen zu erhalten. Ein Zustandsgraph ist ein Petri-Netz, in dem jede Transition

<sup>29</sup>  $P2$  ist in Abbildung 8-1 nicht dargestellt, da die Stelle zur Darstellung der Ereignissequenz ist nicht notwendig

lediglich eine Vorstelle und eine Nachstelle besitzt [ABE90]. Diese Bedingung ist auch durch die Struktur-  
einschränkungen (8-9) und (8-10) beschrieben. Vorteilhaft ist hierbei die Interpretierbarkeit, da sich die  
Anzahl der Marken in einem solchen Petri-Netz nicht ändern kann [ABE90].

Zunächst wird nach Stellen gesucht, die Vorstellen derselben Menge an Transitionen sind. Hierfür muss  
folgende Bedingung erfüllt sein:

$$\exists i, j : (c_{ik} = -1 \wedge c_{jk} = -1, \forall k) \tag{8-24}$$

Mit:

$c_{ik}$ : Element der i-ten Zeile und k-ten Spalte der Inzidenzmatrix  $C = (c_{ik})$

Diese werden dann zusammengeführt, indem die 1-Einträge einer Zeile der Inzidenzmatrix auf die ande-  
re Zeile übertragen werden:

$$c_{ik} = \begin{cases} 1, & \text{wenn } c_{jk} = 1 \\ c_{ik}, & \text{sonst} \end{cases}, \quad \forall k \tag{8-25}$$

Im Anschluss wird die übertragene Zeile (sprich Stelle) gelöscht:

$$c_{jk} = \{\emptyset\}, \quad \forall k \tag{8-26}$$

Dieser Vorgang wird solange wiederholt, bis Bedingung (8-24) für kein Spaltenpaar mehr erfüllt ist.

Um Transitionen zu duplizieren, werden zunächst alle Transitionen gesucht, die mehr als eine Vorstelle  
besitzen:

$$\exists k : |\{(i, j) : (c_{ik} = -1 \wedge c_{jk} = -1)\}| \geq 1 \tag{8-27}$$

Ist eine Transition gefunden, die Bedingung (8-27) erfüllt, wird diese so häufig dupliziert, bis genauso  
viele Transitionen wie Vorstellen existieren. Jeder Transition wird dasselbe (kombinierte) Ereignis anno-  
tiert, wie der Originalen. Im Anschluss wird jeder Transition exakt eine Vorstelle zugewiesen, so dass  
jede Vorstelle einer anderen Transition zugewiesen wurde.

Das Zusammenfassen von Stellen ist im oberen Teil und das Duplizieren von Transitionen im unteren Teil  
von Abbildung 8-2 dargestellt.

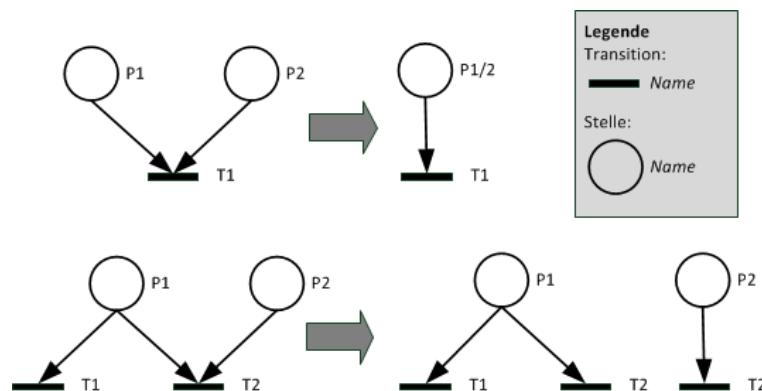


Abbildung 8-2: Zusammenfassen von Stellen (oben) und Duplizieren von Transitionen (unten)

#### 4. Bestimmung der Initialmarkierung

Die Initialmarkierung wird so bestimmt, dass diejenige Transition aktiviert ist, an der das erste Ereignis  $\tilde{e}_1^r(k)$  der Ereignissequenz  $E^r$  annotiert ist. Dies ist exakt die Transition, die durch die erste Spalte von  $\mathcal{C}$  repräsentiert wird, wenn in Schritt 1 das erste Ereignis in  $E^r$  der ersten Zeile und ersten Spalte von  $B(E^r)$  zugewiesen wurde. Ist diese Transition nicht Teil eines Zyklus<sup>30</sup>, so hat der Algorithmus keine Vorstelle für diese Transition generiert. In diesem Fall muss eine neue Stelle (Zeile der Inzidenzmatrix) als Vorstelle der Transition (-1 in der ersten Spalte der Inzidenzmatrix) generiert und mit der Anfangsmarkierung versehen werden. Mit diesem Schritt ist der Algorithmus nach [LELE11] abgeschlossen und die Struktur des auf Maschinenverhalten bezogenen Teils des MSPN generiert. Alle folgenden Schritte der MSPN-Generierung entsprechen Ergänzungen des Algorithmus, die im Rahmen dieser Arbeit erarbeitet wurden.

Im Anschluss an die Strukturgenerierung nach [LELE11] wird die Annotation des 5-Tupels  $\langle d_{min}, d_{max}, \mu, \sigma, n \rangle$  an jede Transition vorgenommen. Zunächst muss jedem Ereignis in der Ereignissequenz eine Transition zugewiesen werden. Hierbei ist zu beachten, dass aufgrund der Duplikation von Transitionen in Schritt 3 des obigen Algorithmus mehrere Transitionen mit demselben Ereignis annotiert sein können. Um zu unterscheiden welche Ereignisse welcher duplizierten Transition zugewiesen werden müssen, werden die vorangegangenen Ereignisse berücksichtigt. Zunächst wird ermittelt, welche Vorstelle die Transitionen besitzen. Daraufhin wird bestimmt, welche Transitionen die ermittelte Stelle im Nachbereich haben und welche Ereignisse an diesen annotiert sind. Somit ist das „Vorereignis“ bekannt. Ereignisse in der Sequenz, die das gleiche Vorereignis wie die entsprechende Transition besitzen, werden dieser zugewiesen. Dieses Vorgehen wird durch das folgende Beispiel verdeutlicht:

Die in Tabelle 8-1 gezeigte Ereignissequenz wurde genutzt um obigen Algorithmus zur Generierung eines Petri-Netzes anzuwenden. Das resultierende Petri-Netz ist in Abbildung 8-3 dargestellt. Zu beachten ist, dass sowohl Transition T2 als auch T4 mit dem Ereignis „C“ annotiert sind. Dies ist das Resultat der Duplikation aus Schritt 3 des Algorithmus.

**Tabelle 8-1: Beispielhafte Ereignissequenz**

Zeitstempel/ms	Ereignis
100	A
300	C
500	D
700	B
900	C
1100	D

Zur Berechnung des zu annotierenden 5-Tupels einer jeden Transition, muss nun ermittelt werden welches Auftreten von Ereignis „C“ welcher Transition zuzuweisen ist. Transition T4 hat als Vorstelle P2, welche wiederum Nachstelle von T3 ist. Diese ist mit dem Ereignis „B“ annotiert. Somit sind alle Vorkommnisse von „C“, dessen Prä-Ereignis „B“ ist, der Transition T4 zuzuweisen. Dies gilt hier für das Ereignis zum Zeitpunkt 900 ms in Tabelle 8-1. Analog dazu sind alle Ereignisse „C“ die als Folge des Ereignisses „A“ auftreten, Transition T2 zuzuweisen. Dies ist der Fall für das Ereignis zum Zeitpunkt 300 ms.

<sup>30</sup> Z.B. weil die Ereignisse der Transition Teil eines Hochfahrvorganges sind

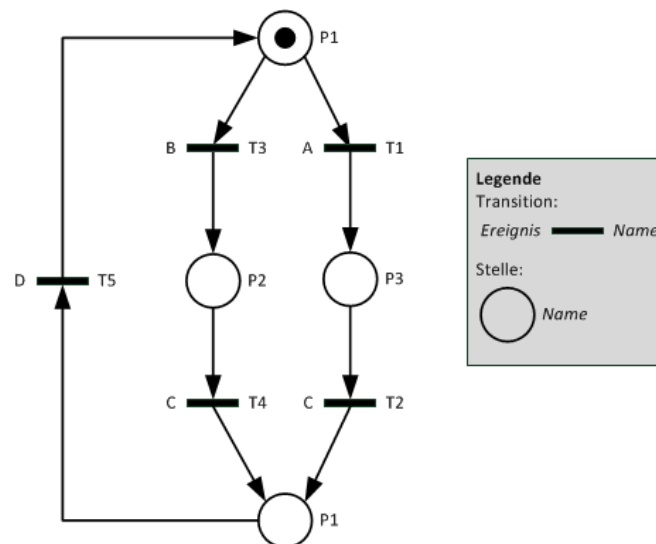


Abbildung 8-3: Beispiel Petri-Netz

Dieses Vorgehen ist in Listing 8-1 als Pseudo-Code angegeben. Im Anschluss an die Zuweisung von Ereignissen zu Transitionen werden die Zeitdifferenzen von allen Ereignissen, die einer Transition zugewiesen sind, zu dessen Vorereignissen ermittelt. Aus den Zeitdifferenzen werden je das Minimum, das Maximum, der Mittelwert, die Standardabweichung sowie deren Anzahl bestimmt und als  $(d_{min}, d_{max}, \mu, \sigma, n)$  annotiert.

```

1 für alle unterschiedlichen Ereignisse  $\tilde{e}_i^{dist}$  in  $E^r$ 
2   für alle Transitionen  $\mathcal{T}_t$  annotiert mit  $\tilde{e}_i^{dist}$ 
3     finde  $\mathcal{S}_s \in \bullet \mathcal{T}_t$ 
4      $preTrans_t = \{ \mathcal{T}_j \in \mathcal{T} : \mathcal{S}_s \in \mathcal{T}_j \bullet \}$ 
5   ende
6   für alle  $\tilde{e}_j^r == \tilde{e}_i^{dist}$ 
7     wenn  $\tilde{e}_{j-1}^r \in \tilde{e}(preTrans_t)$ 
8        $\tilde{e}(\mathcal{T}_t) \leftrightarrow \tilde{e}_j^r$ 
9     ende
10  ende
11 ende
    
```

Listing 8-1: Pseudo Code zur Zuweisung von Ereignissen zu Transitionen

### 8.3.2 Generieren des auf Signalverhalten bezogenen Teils

Das bis hierhin generierte Petri-Netz entspricht dem Maschinenverhalten bezogenen Teil  $\langle \mathcal{S}_{mach}, \mathcal{T}, \mathcal{F}_{mach} \rangle$  eines MSPN. Allerdings ist nicht gewährleistet, dass das Petri-Netz ausschließlich die beobachtete Sprache des Systems reflektiert. Das heißt, dass u.U. auch unbeobachtete Ereignissequenzen durch das Petri-Netz abgebildet werden. Das Netz ist demnach zu permissiv. Der Forderung nach wenig Permissivität steht jedoch die Forderung nach einer Generalisierung des beobachteten Verhaltens gegenüber [LELE11]. Ein minimal-permissives Modell würde ausschließlich die eine beobachtete Sequenz reflektieren, was entsprechend fehlender Generalisierung des Modells zuzuschreiben wäre [LELE11]. Ziel dieses Teils des Algorithmus ist es daher, die Permissivität zu verringern ohne dabei zu stark an Generalität des Modells zu verlieren. Um dies zu erreichen, wird die Tatsache ausgenutzt, dass binäre Signale stets nur zwischen ihren beiden Elementarzuständen wechseln können.

Zunächst werden die Stellen  $\mathcal{S}_{sig}$  generiert. Die Anzahl an Stellen entspricht dabei der doppelten Anzahl an Signalen in der Ereignissequenz  $E^r$  bzw. der technischen Ressource zugeordneten Signalen:

$$|\mathcal{S}_{sig}| = 2 \cdot |\{S_i, \forall i : R(S_i) = R_r\}| \quad (8-28)$$

Jede Stelle aus  $\mathcal{S}_{sig}$  repräsentiert einen 1-Zustand oder 0-Zustand eines Signals. Entsprechend wird jede Stelle  $\mathcal{S}_{(2i-1)}^{sig} \in \mathcal{S}_{sig}$  mit dem 1-Zustand des Signals  $s_i^{high}$  assoziiert und annotiert und jede Stelle  $\mathcal{S}_{(2i)}^{sig} \in \mathcal{S}_{sig}$  mit dem 0-Zustand desselben Signals  $s_i^{low}$ . Im Anschluss werden die Flussrelationen  $\mathcal{F}_{sig}$  zwischen den Transitionen und  $\mathcal{S}_{sig}$  generiert. Für jede Transition, die eine fallende Flanke eines bestimmten Signals als Ereignis annotiert hat, wird eine Kante von der Stelle, die den 1-Zustand des Signals repräsentiert, auf die Transition generiert. Somit ist die Stelle nun Teil des Vorbereiches der Transition. Zusätzlich wird eine Kante von der Transition auf diejenige Stelle generiert, die den 0-Zustand des Signals repräsentiert. Somit ist diese Stelle Teil des Nachbereiches der Transition. Dieses Vorgehen wird durch den Pseudo-Code in Listing 8-2 beschrieben.

Zum Abschluss wird die Initialmarkierung der Stellen  $\mathcal{S}_{sig}$  bestimmt. Sie wird so gesetzt, dass jedes erste Ereignis eines jeden Signals in  $E^r$  eintreten kann. Das heißt, ist das erste Ereignis eines Signals in  $E^r$  eine fallende Flanke, so wird die Stelle, die den 1-Zustand des Signals repräsentiert, mit einer Marke initialisiert. Ist das erste Ereignis eines Signals in  $E^r$  eine steigende Flanke, so wird die Stelle, die den 0-Zustand des Signals repräsentiert, mit einer Marke initialisiert. Somit ist gewährleistet, dass das MSPN denselben Initialzustand hat wie die technische Ressource zu Beginn der Beobachtung.

```

1  für alle Transitionen  $\mathcal{T}_t$  in  $\mathcal{T}$ 
2    wähle  $\tilde{e}_i^r$  annotiert an  $\mathcal{T}_t$ 
3    für alle Ereignisse  $e_j^r$  innerhalb von  $\tilde{e}_i^r$ 
4      wenn  $e_j^r == s_j^\uparrow$ 
5         $\mathcal{S}_{(2j-1)}^{sig} \rightarrow \bullet \mathcal{T}_t$ 
6         $\mathcal{S}_{2j}^{sig} \rightarrow \mathcal{T}_t \bullet$ 
7      wenn sonst  $e_j^r == s_j^\downarrow$ 
8         $\mathcal{S}_{(2j-1)}^{sig} \rightarrow \mathcal{T}_t \bullet$ 
9         $\mathcal{S}_{2j}^{sig} \rightarrow \bullet \mathcal{T}_t$ 
10     ende
11   ende
12 ende
    
```

Listing 8-2: Pseudo Code zur Generierung der auf Signalverhalten bezogenen Flussrelationen

## 8.4 Beispiel

Zum besseren Verständnis des Algorithmus' wird hier ein Beispiel aus der in Abschnitt 11.2 vorgestellten Fallstudie erläutert. Das in Abbildung 8-4 gezeigte Förderband ist eine in der Fallstudienanlage enthaltene technische Ressource.

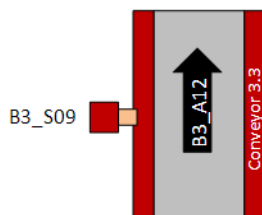


Abbildung 8-4: Förderband der Fallstudie

Das Förderband *Conveyor 3.3* enthält einen Näherungssensor *B3\_S09* zur Erkennung von Werkstücken und einen Motorantrieb, der über das Signal *B3\_A12* angesteuert wird. Das Förderband befindet sich vor einer Werkzeugmaschine. Wird das Transportband angefordert, so aktiviert es Aktor *B3\_A12*, bis ein Werkstück von *B3\_S09* erkannt wird. Daraufhin hält das Förderband an, bis die Werkzeugmaschine die Bearbeitung beendet hat. Im Anschluss wird das Werkstück abtransportiert, indem *B3\_A12* erneut aktiviert wird. Im Anschluss wird *B3\_A12* wieder deaktiviert und das Förderband befindet sich im Initialzustand. Die in Tabelle 8-2 dargestellte Ereignissequenz zeigt den Durchlauf von zwei Werkstücken über das Förderband<sup>31</sup>. Die erste Spalte der Tabelle enthält eine laufende Nummer zur Referenzierung und ist nicht Teil der eigentlichen Ereignissequenz.

Tabelle 8-2: Teil der aufgenommenen Ereignissequenz

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis
1	388400	'B3_A12'	1
2	389976	'B3_S09'	1
3	389992	'B3_A12'	0
4	398088	'B3_A12'	1
5	398244	'B3_S09'	0
6	399430	'B3_A12'	0
7	435091	'B3_A12'	1
8	436698	'B3_A12'	0
9	436698	'B3_S09'	1
10	444795	'B3_A12'	1
11	444951	'B3_S09'	0
12	446183	'B3_A12'	0

Zunächst wird die Zeitschwelle  $T_{thresh} = 100\text{ ms}$  festgelegt. Dies entspricht genau einem OPC-Serverzyklus während der Aufnahme der Ereignissequenz. Alle Ereignisse, die innerhalb dieser Zeit gemeinsam auftreten, werden entsprechend Abschnitt 8.2 zusammengeführt. Das Ergebnis ist in Tabelle 8-3 dargestellt. Ereignisse, die Zusammengeführt wurden (z.B. in Zeile 2), erhielten einen neuen Signalnamen, der sich aus den ursprünglichen Namen mit zwischengestelltem „\_AND\_“ zusammensetzt. Die Elementarzustände, die die Signale einnehmen, wurden in Vektoren zusammengefasst. Hierbei sind die Elementarzustände den Signalen in der Reihenfolge zuzuordnen, in der sie im neuen Namen des Ereignisses vorkommen. In Zeile 2, zum Beispiel, nimmt das Signal *B3\_A12* den Zustand 0 ein und *B3\_S09* den Zustand 1.

<sup>31</sup> Die gesamte Ereignissequenz enthält den Durchlauf von 5 Werkstücken und ist in Anhang D zu finden

Tabelle 8-3: Ereignissequenz nach Zusammenführen von Ereignissen innerhalb  $T_{thresh} = 100ms$ 

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis	Ereign. Nr.
1	388400	'B3_A12'	1	1
2	389976	'B3_A12_AND_B3_S09'	[0 1]	2
3	398088	'B3_A12'	1	1
4	398244	'B3_S09'	0	3
5	399430	'B3_A12'	0	4
6	435091	'B3_A12'	1	1
7	436698	'B3_A12_AND_B3_S09'	[0 1]	2
8	444795	'B3_A12'	1	1
9	444951	'B3_S09'	0	3
10	446183	'B3_A12'	0	4

Aus dieser Ereignissequenz wird nun gem. Formel (8-21) die Ereignispropagationsmatrix generiert:

$$B(E^r) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (8-29)$$

Die Zeilen und Spalten sind dabei in dieser Reihenfolge den Ereignissen  $B3\_A12\_high$ ,  $B3\_A12\_low\_AND\_B3\_S09\_high$ ,  $B3\_S09\_low$  und  $B3\_A12\_low$  zugewiesen (siehe auch Nummerierung i der letzten Spalte von Tabelle 8-3). Der Ausdruck „low“ steht hierbei für eine fallende Flanke des Signals und der Ausdruck „high“ für eine steigende Flanke. Aus der Ereignispropagationsmatrix wird die Inzidenzmatrix bestimmt:

$$C = I_v - B(E^r)^T = \begin{pmatrix} 1 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix} \quad (8-30)$$

Grafisch dargestellt und um o.g. Ereignisannotationen ergänzt, entspricht das Petri-Netz dem in Abbildung 8-5 links dargestellten. In Schritt 3 werden Stellen zusammengeführt, die Vorstellen derselben Menge an Transitionen sind (Formeln (8-24) - (8-26)). Dies trifft hier für die Stellen P2 und P4 zu. Daher werden diese zusammengeführt (P2 auf der rechten Seite von Abbildung 8-5). Ein Duplizieren von Transitionen ist hier nicht notwendig, da nach dem Zusammenführen der Stellen keine Transition mehrere eingehende Kanten besitzt. Das erste Ereignis in der Sequenz ist die steigende Flanke von  $B3\_A12$  (siehe Tabelle 8-3). Daher wird P2 mit der Anfangsmarkierung besetzt, so dass die Transition mit dem entsprechenden annotierten Ereignis (hier T1) aktiviert ist. Das resultierende Petri-Netz ist in Abbildung 8-5 rechts dargestellt.

Es ist zu erkennen, dass das Petri-Netz zwar das Verhalten des Transportbandes widerspiegelt, jedoch auch unbeobachtetes Verhalten enthält. So entspricht die Feuersequenz (T1 → T2 → T1 → T2) einer Ereignissequenz, in der zwei steigende Flanken des Signals  $B3\_S09$  enthalten sind, jedoch keine fallende Flanke desselben Signals. Um dieses Verhalten des MSPN zu verhindern, wird im nächsten Schritt der auf Signalverhalten bezogene Teil generiert. Die Ereignissequenz enthält Ereignisse von exakt zwei Signalen. Somit werden vier neue Stellen generiert, die den 1-Zuständen und 0-Zuständen der beiden Signale entsprechen. Enthält eine Transition ein Ereignis einer steigenden Flanke eines Signals, muss die

Marke aus der Stelle, die den 0-Zustand des Signals repräsentiert, entfernt und der Stelle hinzugefügt werden, die den 1-Zustand des Signals repräsentiert.

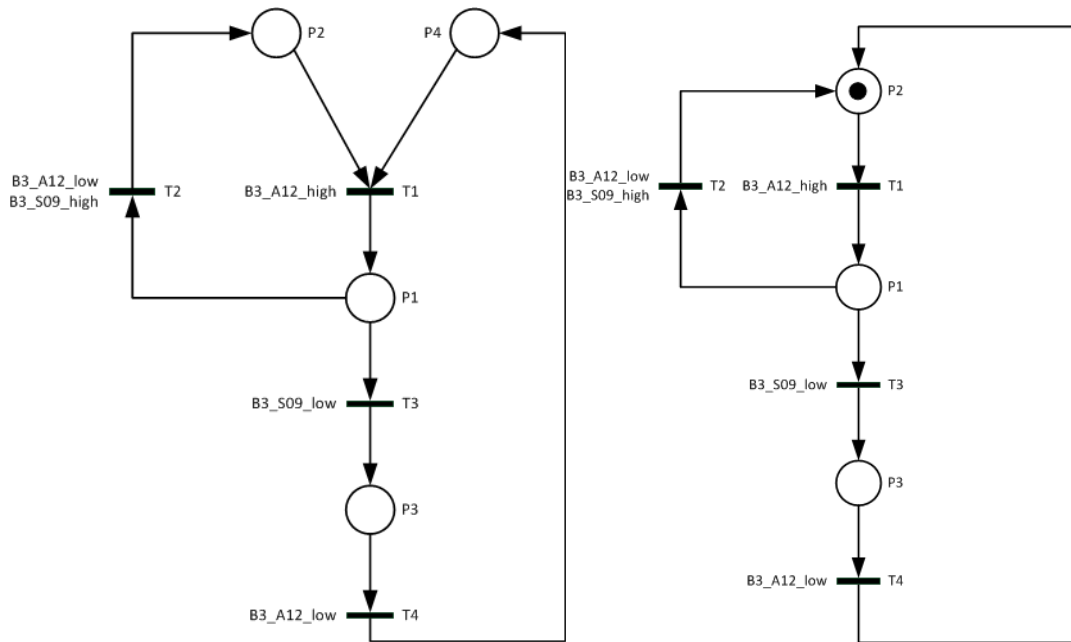


Abbildung 8-5: Petri-Netz nach Durchführung von Schritt 2 (links) sowie Schritt 3 und 4 (rechts)

Da in der Ereignissequenz sowohl das erste Ereignis von *B3\_S09* als auch das erste Ereignis von *B3\_A12* je eine steigende Flanke ist, werden die Stellen, die die 0-Zustände der beiden Signale repräsentieren, mit einer Initialmarkierung besetzt. Daraus resultiert das in Abbildung 8-6 dargestellte Petri-Netz. Der untere Abschnitt der Abbildung zeigt den auf Maschinenverhalten bezogenen Teil und der obere Abschnitt den auf Signalverhalten bezogenen Teil des MSPN. Die Annotationen an den aus Signalverhalten bezogenen Stellen entsprechen dabei den Signalzuständen, die sie repräsentieren. Das resultierende Petri-Netz lässt nun nur noch eine sich wiederholende Feuersequenz zu. Diese ist  $T1 \rightarrow T2 \rightarrow T1 \rightarrow T3 \rightarrow T4$ . Dies entspricht der Ereignissequenz, die zyklisch in Tabelle 8-3 auftritt.

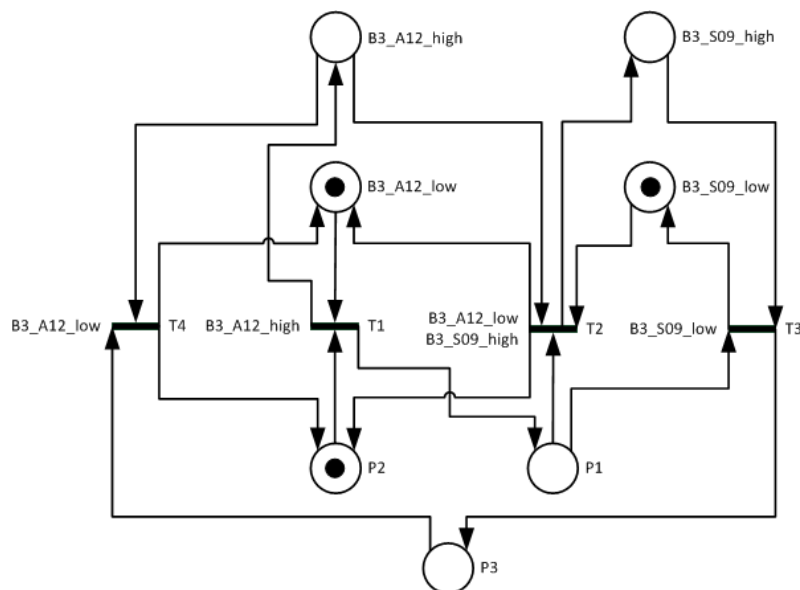


Abbildung 8-6: Finale MSPN-Struktur des Beispiels

Im letzten Teil des Algorithmus wird das 5-Tupel zur Annotation an die Transitionen aus der Ereignissequenz berechnet. Die Berechnung sei beispielhaft anhand von Transition T2 erläutert. T2 hat das kombinierte Ereignis aus der steigenden Flanke von  $B3\_S09$  und der fallenden Flanke von  $B3\_A12$  annotiert. Diese treten in der Ereignissequenz in den Zeilen 2, 7, 12, 17 und 22 auf (siehe Tabelle D-1 im Anhang). Hieraus kann bereits die Anzahl an Ereignissen  $n=5$  bestimmt werden. Die Zeitdifferenzen der Zeitstempel zu den jeweils vorherigen Zeitstempeln (aus den Zeilen 1, 6, 11, 16 und 21) werden berechnet. Hieraus kann nun der Minimalwert und Maximalwert sowie Mittelwert und Standardabweichung bestimmt werden. Das Ergebnis der Berechnung für alle Transitionen ist in Tabelle 8-4 dargestellt. Alle Zeiten sind in Millisekunden angegeben.

**Tabelle 8-4: Berechnete 5-Tupel für alle Transitionen des Beispiels**

	$d_{min}$	$d_{max}$	$\mu$	$\sigma$	$n$
<b>T1</b>	8.097	158.3	43.36	52.58	10
<b>T2</b>	1.529	1.607	1.579	0.032	5
<b>T3</b>	0.141	0.156	0.153	0.007	5
<b>T4</b>	1.186	1.248	1.226	0.0259	5

## 8.5 Anomalieerkennung

Zur Erkennung von Anomalien im Verhalten der Ressourcen als Hinweis auf eine evolutionäre Änderung wird das Verhalten eines aus der Ereignissequenz  $E_{obs}^r$  generierten Maschinenzustandsmodells  $MSPN^r$  mit dem aktuell beobachtbaren Verhalten der Ressource verglichen. Hierzu wird dem MSPN jedes zum Zeitpunkt  $k \cdot T_{OPC}$  auftretende Ereignis  $e_j(k)$  zugeführt, das der Ressource zugewiesen ist. Eine Filterung des gesamten Ereignisstromes der Anlage findet gem. Filterregel (8-17) statt. Entsprechend jedem eingehenden Ereignisses wird, sofern möglich, die Markierung des MSPN  $m_k$  mittels Feuerung einer aktivierten Transition, die mit selbigem Ereignis annotiert ist, neu berechnet. Ist das Ereignis Teil eines im MSPN kombinierten Ereignisses, so wird das Auftreten vorgemerkt, bis alle Ereignisse des kombinierten Ereignisses beobachtet wurden und die Transition feuern kann oder eine Anomalie erkannt wird. Zudem wird für jede Transition die Anzahl an Feuervorgängen während der Anomalieerkennung gezählt und gespeichert.

Tritt ein Ereignis auf, nicht auf oder zu einem Zeitpunkt auf, der dem Verhalten des MSPN widerspricht, wird dies als eine Anomalie gewertet. Aus dieser Aussage können Anomalien abgeleitet werden, die einem MSPN widersprüchliches Verhalten beschreiben. Folgend werden sechs solcher Anomalien vorgestellt, die auf eine evolutionäre Änderung hinweisen:

1. Es tritt ein Ereignis zum Zeitpunkt  $k$  auf, das an keiner Transition annotiert ist:

$$e_j(k) \notin E_{obs}^r \quad (8-31)$$

In diesem Fall muss es sich um ein neues Signal handeln oder um ein Signal, das bisher nicht beobachtet wurde. Ein Beispiel hierfür ist ein neu hinzugefügter Sensor.

2. Es tritt ein Ereignis auf, das an keiner aktivierten Transition annotiert ist:

$$\nexists \mathcal{T}_t \in MSPN^r : \mathcal{T}_t \text{ aktiviert} \wedge (e_j(k) \subseteq \tilde{e}(\mathcal{T}_t)) \quad (8-32)$$

Diese Anomalie deutet auf eine geänderte Kausalfolge von Ereignissen hin. Diese Anomalie kann bspw. auf geänderte Ablaufsequenzen in der Automatisierungssoftware hinweisen, oder auch auf physikalisch-strukturelle Änderungen an der technischen Ressource.

3. Es tritt ein Ereignis  $e_j(k)$  auf und es existiert keine aktivierte Transition, an der  $e_j(k)$  und eine Mindestaktivierungsdauer  $d_{min}$  annotiert ist, die kleiner der Differenz  $(k - k') \cdot T_{OPC}$  zwischen den Zeitstempeln von  $e_j(k)$  und des vorangegangenen Ereignisses  $e_{j-1}(k')$  ist:

$$\nexists \mathcal{T}_t \in MSPN^r : \mathcal{T}_t \text{ aktiviert} \wedge (e_j(k) \subseteq \check{e}(\mathcal{T}_t)) \wedge ((k - k') \cdot T_{OPC} > d_{min}(\mathcal{T}_t)) \quad (8-33)$$

In diesem Fall hat sich das zeitliche Verhalten der Ressource verändert. Mindestens ein Ereignis tritt in einem geringeren zeitlichen Abstand zum vorigen Ereignis auf als zuvor. Diese Anomalie kann einen Hinweis auf schnellere Abläufe der Ressource geben.

4. Die Differenz der aktuellen Zeit  $k \cdot T_{OPC}$  und eines vorangegangenen Ereignisses  $e_{j-n}(k')$  ist größer als die annotierte Maximalaktivierungsdauer  $d_{max}$  aller aktivierten Transitionen, an denen  $e_{j-n}(k')$  annotiert ist:

$$(k - k') \cdot T_{OPC} > d_{max}(\mathcal{T}_t), \forall \mathcal{T}_t \in MSPN^r : \mathcal{T}_t \text{ aktiviert} \wedge (e_{j-n}(k') \subseteq \check{e}(\mathcal{T}_t)) \quad (8-34)$$

In diesem Fall hat sich das zeitliche Verhalten der Ressource verändert, da mindestens ein Ereignis später als im Beobachtungszeitraum vor der Generierung des MSPN auftritt. Somit kann es sich bei einer solchen Anomalie um einen verlangsamten Ablauf handeln, der z.B. aufgrund von Verschleiß auftreten könnte. Es sei zu beachten, dass in einem MSPN nur eine Transition aktiviert sein kann, aufgrund der Einschränkung (8-9) und (8-10). Somit genügt es stets die Differenz der Zeitstempel zwischen dem aktuellen und dem vorangegangenen Ereignis zu prüfen.

5. Es ist ein Ereignis aufgetreten, das zu einem an einer aktivierten Transition annotierten kombinierten Ereignis  $\check{e}_j^f$  gehört, jedoch tritt mindestens ein weiteres zum kombinierten Ereignis zugehöriges Ereignis nicht innerhalb der definierten Zeitschwelle  $T_{thresh}$  auf:

$$\exists x, y \in \mathbb{N} : (e_{j-x}(k - y) \subseteq \check{e}(\mathcal{T}_t)) \wedge \mathcal{T}_t \text{ aktiviert} \wedge y \cdot T_{OPC} \geq T_{thresh} \quad (8-35)$$

Diese Anomalie zeigt auf, dass Ereignisse, die zuvor stets gemeinsam in der gegebenen Zeitschwelle auftraten, dies nun nicht mehr tun. Somit hat sich das kausale zeitliche Verhalten geändert. Auch hier könnte eine verlangsamte Abfolge aufgrund von Verschleiß die Ursache sein. Bei dieser Anomalie sei zu beachten, dass ebenfalls die Zeitschwelle zu gering gewählt sein könnte.

6. Für Stellen mit mehr als einer Nachtransition ist das Verhältnis der Anzahl an Feuervorgängen der beiden Transitionen während des Zeitraumes der Anomalieerkennung sehr viel kleiner oder sehr viel größer als das Verhältnis der annotierten Anzahl an Feuervorgängen selbiger Transitionen. Die maximal erlaubte Abweichung kann mithilfe eines Sensitivitätsmaßes angegeben werden.

$$\exists \{x, y\} \in \mathbb{N}, \mathcal{S} \in \bullet \mathcal{T}_x \cap \bullet \mathcal{T}_y : \left| \frac{n(\mathcal{T}_x)}{n(\mathcal{T}_y)} - \frac{n'(\mathcal{T}_x, k)}{n'(\mathcal{T}_y, k)} \right| \geq \Delta \quad (8-36)$$

Mit:

$n'(T_t, k)$ : Anzahl an Feuervorgängen der Transition  $T_t$  während der Anomalieerkennung zum Zeitpunkt  $k$

$n(T_t)$ : Annotierte Anzahl an Feuervorgängen der Transition  $T_t$

$\Delta$ : Sensitivitätsmaß zur Erkennung nicht mehr vorhandenen Verhaltens

Überschreitet das Verhältnis die durch das Sensitivitätsmaß  $\Delta$  vorgegebene Abweichung, so ist dies ein Hinweis darauf, dass ein im MSPN vorhandenes Verhalten nun nicht mehr Teil des Verhaltens der zugehörigen Ressource ist. Zum Beispiel könnte eine entsprechende Funktion der Ressource geändert worden sein. Diese Anomalie kann jedoch stets nur einen Hinweis auf nicht vorhandenes Verhalten geben.

## 9 Materialflussmodelle: Definition, Lernalgorithmus und Anomalieerkennung

Im Bereich der Kennzahlen wird der Logistik und dem Materialfluss eine besondere Wichtigkeit zugeschrieben. Dies zeigt sich auch an der Vielzahl der Richtlinien, die sich mit Kennzahlen für die (Intra-) Logistik [VDI 2525; VDI 4400-2; VDI 4490; VDI 4486], konkret für den Materialfluss [VDI 2689] oder das Fördersystem [VDI 3649; VDI 3978] von Fertigungssystemen beschäftigen. Auch die vorangegangenen Untersuchungen in Kapitel 3 haben gezeigt, dass ein Großteil der Merkmale sich aus den Transportvorgängen bzw. der Verweildauer von Werkstücken in technischen Ressourcen oder der gesamten Anlage ableiten lässt<sup>32</sup>. Transportvorgänge machen zudem i.d.R. einen größeren Teil der Produktionszeit in der Fertigung aus als die eigentliche Bearbeitung [GRO14]. Aus diesem Grund gilt der Materialfluss als eine der wichtigsten Abbildungen zur Bewertung von Fertigungssystemen [LAN99; BESÄ10]. Vereinzelt Ansätze, wie [SWV10; WVS+10] haben sich zwar bereits mit der auf Daten basierten Generierung von Materialflussmodellen beschäftigt, gehen allerdings von einer anderen Datengrundlage aus als die vorliegende Arbeit (vgl. Abschnitt 5.2). Die in [SWV10; WVS+10] zugrundeliegende Annahme ist, dass zu jedem Ereignis der Ort, die Zeit sowie die ID des auslösenden Objektes bekannt sind. Ähnliches gilt für die Anwendung von Process Discovery Algorithmen aus dem Bereich des Process Mining, u.a. [VAN11; HLW06], die ebenfalls davon ausgehen, dass mindestens bekannt ist, welche Prozessinstanzen die Ereignisse ausgelöst haben (vgl. Abschnitt 5.2). Im Falle der Process Discovery Algorithmen, haben Untersuchungen in [FÜL14<sup>6</sup>] gezeigt, dass selbst wenn die nötigen Informationen vorhanden sind, die resultierenden Modelle einen hohen Komplexitätsgrad aufweisen und nicht geeignet sind für eine Untersuchung hinsichtlich nicht-funktionaler Eigenschaften von Fertigungssystemen, wie sie in Abschnitt 3.3 diskutiert wurden. Aus diesem Grund wurden im Rahmen dieser Arbeit ein neuer Modelltyp sowie ein zugehöriger Lernalgorithmus erarbeitet. Ziel war dabei stets, möglichst simple und damit gut zur Dokumentation geeignete Modelle zu generieren, die für eine Analyse hinsichtlich aus dem Materialfluss resultierender Qualitätsmerkmale geeignet sind.

MSPN geben den Materialfluss im Fertigungssystem nicht explizit wieder. Um für den Materialfluss relevante Eigenschaften analysieren zu können stellt dieser Abschnitt Aspektmodelle vor, die den Materialfluss innerhalb des Fertigungssystems abbilden (Abschnitt 9.1). Im Gegensatz zu MSPN stellen sie nicht das Verhalten einzelner technischer Ressourcen unter Berücksichtigung aller zugehörigen relevanten Signale dar. Vielmehr bedienen sie sich nur einer für den Materialfluss relevanten Untermenge der Signale und stellen den Materialfluss ressourcenübergreifend dar. Im Anschluss an die Definition der Aspekt-Modelle werden der Lernalgorithmus zur Generierung dieser (Abschnitte 9.2 und 9.3) sowie zugehörige Anomalien (Abschnitt 9.4) vorgestellt. Da auch die Materialflussmodelle auf Petri-Netzen basieren, werden sie im Folgenden *Material Flow Petri Nets (MFPN)* genannt. Entgegen dem vorigen Kapitel wird hier ein laufendes Beispiel eingesetzt, das gemeinsam mit den erläuterten Schritten des Algorithmus fortgeführt wird.

### 9.1 Definition

Ein MFPN ist ein Petri-Netz  $\mathcal{PN} = \langle \mathcal{S}, \mathcal{T}, \mathcal{F}, m \rangle$ , mit einer Menge an Stellen  $\mathcal{S}$ , einer Menge an Transitionen  $\mathcal{T}$  und einer Menge an Flussrelationen  $\mathcal{F} \subseteq (\mathcal{S} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{S})$  und einer Menge Marken  $m$ , die den Stellen zugeordnet sind. Der Vorbereich und Nachbereich einer Transition sowie deren Kardina-

<sup>32</sup> Vgl. auch Tabelle C-1 im Anhang

litäten sind analog zu den Transitionen eines MSPN definiert, siehe (8-1) und (8-2). Zudem entspricht der *Vorbereich*  $\bullet\mathcal{S}_s$  einer Stelle denjenigen Transitionen, von denen aus eine Kante auf die Stelle zeigt:

$$\bullet\mathcal{S}_s = \{ \mathcal{T} \in \mathcal{T} : (\mathcal{T}, \mathcal{S}_s) \in \mathcal{F} \} \quad (9-1)$$

Analog dazu entsprechen diejenigen Transitionen, auf die eine von  $\mathcal{S}_s$  ausgehende Kante zeigt, dem *Nachbereich*  $\mathcal{S}_s\bullet$  der Stelle:

$$\mathcal{S}_s\bullet = \{ \mathcal{T} \in \mathcal{T} : (\mathcal{S}_s, \mathcal{T}) \in \mathcal{F} \} \quad (9-2)$$

Der Term  $|\bullet\mathcal{S}_s(\mathcal{T})|$  beschreibt die Kardinalität des Vorbereiches einer Stelle, also die Anzahl an Vortransitionen. Analog dazu beschreibt der Term  $|\mathcal{S}_s\bullet(\mathcal{T})|$  die Kardinalität des Nachbereiches.

Eine Marke in einem MFPN ist als ein Werkstück im Transportsystem zu interpretieren. Jeder Marke ist ein Werkstücktyp aus der endlichen Menge  $\mathbf{W}$  zugewiesen:

$$W(m) = f: \mathbf{W} \rightarrow m \quad (9-3)$$

Dies entspricht dem Prinzip der *farbigen Petri-Netze*, vgl. [DAAL10]. Die Farbe der Marke ist entsprechend als *Werkstücktyp* zu interpretieren.

Für die Struktur eines MFPN gelten die folgenden strukturellen Einschränkungen:

$$|\bullet\mathcal{T}_t(\mathcal{S})| \leq 1, \forall t \quad (9-4)$$

$$|\mathcal{T}_t\bullet(\mathcal{S})| = 1, \forall t \quad (9-5)$$

$$|\bullet\mathcal{S}_s(\mathcal{T})| > 1 \Leftrightarrow |\mathcal{S}_s\bullet(\mathcal{T})| > 1, \forall s \quad (9-6)$$

Die Einschränkungen (9-4) und (9-5) implizieren, dass jede Transition stets maximal eine Vorstelle besitzt und genau eine Nachstelle besitzt. Einschränkung (9-6) sagt aus, dass eine Stelle maximal eine Nachtransition besitzen kann, wenn sie mehr als eine Vortransition besitzt. Andersherum kann die Stelle nur maximal eine Vortransition besitzen, wenn sie mehr als eine Nachtransition besitzt.

Jede Transition ist mit einem (kombinierten) Signal  $\tilde{\mathcal{S}}(\mathcal{T}_t)$  annotiert. Dieses Signal entspricht einer detektierbaren Position eines Werkstückes. Zudem ist jede Transition mit einem Vektor von 6-Tupeln  $(d_{min}, d_{max}, \mu, \sigma, n, \mathbf{ident})$  annotiert. Die Elemente der 6-Tupel sind wie folgt definiert:

$d_{min}$ : Minimale Aufenthaltsdauer an detektierter Position

$d_{max}$ : Maximale Aufenthaltsdauer an detektierter Position

$\mu$ : Mittlere Aufenthaltsdauer an detektierter Position

$\sigma$ : Standardabweichung der Aufenthaltsdauer an detektierter Position

$n$ : Anzahl an Transportvorgängen an detektierter Position

**ident**: Vektor aus Elementarzuständen von Werkstück identifizierenden Signalen

Der Vektor **ident** beschreibt einen Werkstücktypen anhand einer Menge von Elementarzuständen von Sensoren, die Werkstücktypen identifizieren können (z.B. Induktive Sensoren zur Unterscheidung zwischen metallischen und nicht metallischen Werkstücken). Somit ist jedem **ident**-Vektor ein Werkstücktyp zuzuweisen, und er codiert die Farbe einer Marke. Für jeden Werkstücktypen, der die Position passiert, die durch die Transition repräsentiert wird, wird ein 6-Tupel annotiert. Mit anderen Worten ent-

hält jedes 6-Tupel einer Transition einen anderen *ident*-Vektor. Somit kann in einer Analyse zwischen den Materialflüssen unterschiedlicher Werkstücktypen unterschieden werden. Die ersten vier Elemente  $d_{min}, d_{max}, \mu, \sigma$  des 6-Tupels beziehen sich auf die Dauern, die der Zustand von  $\tilde{S}(T_t)$  gleich 1 ist, d.h. die Dauer zwischen der steigenden Flanke der detektierten Position und der fallenden Flanke der detektierten Position des entsprechenden Werkstücktypen. Der Wert  $n$  bezieht sich auf die Anzahl an Werkstücken des entsprechenden Typs, die die Position passiert haben.

Stellen können ebenfalls mit den 6-Tupel-Vektoren annotiert sein. Allerdings beziehen sich die Zeiten hier anstatt auf Verweildauern auf die Transportdauer zwischen zwei detektierten Positionen. Das heißt, sie beziehen sich auf die Zeitdifferenzen zwischen der fallenden Flanke des an der Vortransition annotierten Signals und der steigenden Flanke des an der Nachtransition annotierten Signals. Besitzt eine Stelle keine Nachtransition, ist sie nicht annotiert.

Folgendes Beispiel soll dies verdeutlichen: Eine Transportstrecke enthält zwei Lichtschranken, die die Signale *S1* und *S2* auslösen. Zwei unterschiedliche Werkstücktypen werden hier transportiert. Diese werden durch die Vektoren *ident1* und *ident2* beschrieben. Das in Abbildung 9-1 dargestellte MFPN zeigt diese Transportstrecke und wurde aus einem Lauf generiert, in dem vier Werkstücke des Typen *ident1* und drei Werkstücke des Typen *ident2* transportiert wurden. Die obere Transition entspricht dabei der Position an Sensor *S1* und die untere an *S2*. Es sind jeweils zwei 6-Tupel annotiert, je einer für *ident1* und einer für *ident2*. Die Zeiten (hier in Sekunden angegeben) beziehen sich auf die Auslösedauer der Sensoren, wenn die Werkstücke von ihnen detektiert wurden. Auf die Transportdauer zwischen den Sensoren beziehen sich die Annotationen an P1.

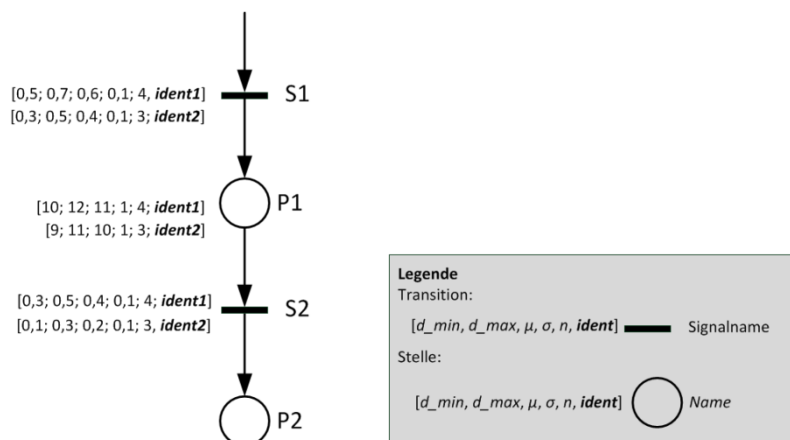


Abbildung 9-1: Beispiel eines MFPN

Hat eine Stelle mehr als eine Nachtransition (d.h. mehr als eine ausgehende Kante), so ist die Stelle selbst nicht annotiert, sondern jeweils ihre ausgehenden Kanten. Die Annotationen beziehen sich dann jeweils auf die Transportdauer zwischen dem Sensor, der an der Vortransition annotiert ist, und dem Sensor, der an der Transition annotiert ist, auf die die Kante zeigt. Dadurch können Transportdauern zwischen der Vortransition und jeder Nachtransition unterschieden werden.

Dies soll ebenfalls anhand eines Beispiels verdeutlicht werden (Abbildung 9-2). Hier existiert nur ein Werkstücktyp, dieser kann jedoch zwei Wege im Transportsystem nehmen. Der erste Weg führt vorbei an *S1* und *S2*. Der zweite Weg wird mittels *S1* und *S3* detektiert. Zur Analyse ist zwischen den Transportwegen von *S1* zu *S2* und von *S1* zu *S3* zu unterscheiden. Dies kann nicht geschehen, wenn lediglich P1 annotiert ist. Daher wird die Transportdauer zwischen *S1* und *S2* an der Kante von P1 zur Transition *S2* annotiert. Die Transportdauer zwischen *S1* und *S3* wird an der Kante zwischen P1 und der Transition

S3 annotiert. Analog dazu wird auch mit Stellen verfahren, die mehrere Vortransitionen besitzen. In diesem Fall werden die eingehenden Kanten der Stelle annotiert.

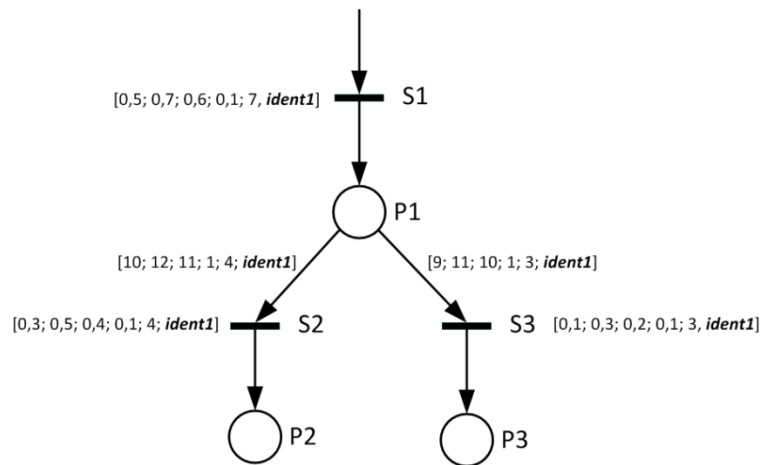


Abbildung 9-2: Beispiel eines MFPN mit einer Stelle, die mehrere Nachtransitionen besitzt

Anhand eines Negativ-Beispiels soll hier auch erläutert werden, warum Stellen nicht mehrere Vortransitionen und mehrere Nachtransitionen besitzen dürfen (Einschränkung (9-6)). Man nehme an, ein weiterer Sensor  $S4$  existiert in dem obigen Beispiel und es wurden vier verschiedene Wege beobachtet ( $S1 \rightarrow S2$ ,  $S1 \rightarrow S3$ ,  $S4 \rightarrow S2$  und  $S4 \rightarrow S3$ ). Ein MFPN ohne Einschränkung (9-6) könnte dann wie in Abbildung 9-3 dargestellt aussehen.

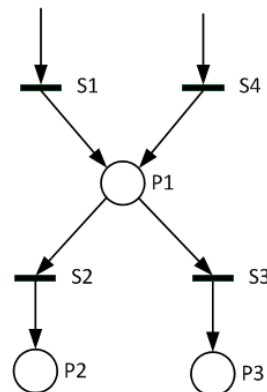


Abbildung 9-3: Mehrdeutiges MFPN (annotierte 6-Tupel nicht dargestellt)

Aus diesem Petri-Netz könnten die Zeitannotation nicht eindeutig unterschieden werden. Konkret heißt dies, dass u.a. einer Annotation an der Kante zwischen P1 und S2 nicht eindeutig einer der Wege  $S1 \rightarrow S2$  oder  $S4 \rightarrow S2$  zugewiesen werden könnte.

Eine Transition wird als *aktiviert* bezeichnet, wenn die folgenden Bedingungen erfüllt sind:

- 1) Die Vorstelle der Transition ist mit einer Marke besetzt oder die Transition besitzt keine Vorstelle:

$$(\mathcal{S} \in \bullet \mathcal{T}_t : m(\mathcal{S}) = 1) \vee (\bullet \mathcal{T}_t = \{\emptyset\}) \tag{9-7}$$

- 2) In der Vorstelle der Transition befindet sich eine Marke deren Farbe einem annotierten **ident**-Vektor der Transition entspricht:

$$\exists \mathbf{ident} \in \text{Annot}(\mathcal{T}_t), \mathcal{S} \in \bullet\mathcal{T}_t : W(m(\mathcal{S})) \rightarrow \mathbf{ident} \quad (9-8)$$

Mit:

$\text{Annot}(\mathcal{T}_t)$ : Menge aller an Transition  $\mathcal{T}_t$  annotierten Elemente

Der Vorgang eine Marke aus der Vorstelle  $\mathcal{S}'$  einer Transition zu entfernen und in die Nachstelle  $\mathcal{S}$  der Transition zu transferieren, wird als *feuern* bezeichnet. Eine Transition  $\mathcal{T}_t$  eines MFPN feuert, wenn die folgenden Bedingungen erfüllt sind:

- (1) Die Transition ist aktiviert,
- (2) Die Aktivierungsdauer liegt zwischen  $d_{min}$  und  $d_{max}$  annotiert für den entsprechenden Werkstücktypen an der Vorstelle bzw. der eingehenden Kante der Transition. Befinden sich mehrere Marken in der Vorstelle, so muss die Verweildauer der Marken anstelle der Aktivierungsdauer berücksichtigt werden,
- (3) Eine steigende Flanke des an der Transition annotierten Signals tritt auf.

Befinden sich in einer Vorstelle einer aktivierten Transition mehrere Marken gleichen Werkstücktyps, so wird stets diejenige Marke transferiert, die die Stelle als erstes erreicht hat. Der Feuervorgang ist dabei nicht als zeitloses Ereignis zu verstehen, sondern benötigt eine Dauer. Diese Dauer liegt zwischen  $d_{min}$  und  $d_{max}$  annotiert an der Transition für den entsprechenden Werkstücktypen. Der Feuervorgang ist abgeschlossen, wenn die fallende Flanke des annotierten Signals auftritt. Eine Transition, die keine Vorstelle besitzt, generiert beim Feuern eine neue Marke. Diese Transition entspricht einem Eingang in das Materialflusssystem.

## 9.2 Preprocessing der Ereignissequenz

Der Materialfluss in Fertigungssystemen ist üblicherweise stark nebenläufig. Dies liegt darin begründet, dass sich i.d.R. mehrere Werkstücke zur selben Zeit in den Anlagen befinden. Deren Transportvorgänge finden weitestgehend unabhängig voneinander, jedoch zeitgleich, statt. Daher ist es wünschenswert, die Materialflüsse der einzelnen Werkstücke zunächst unabhängig voneinander zu betrachten. Durch den zeitgleichen Ablauf der verschiedenen Transportprozesse sind die Nebenläufigkeiten jedoch auch in den resultierenden Ereignissequenzen enthalten. Diese Nebenläufigkeiten müssen aus den Sequenzen eliminiert werden, um eine differenzierte Darstellung der einzelnen Transportvorgänge im System zu ermöglichen. Zudem sind ggf. Teile der für den Materialfluss relevanten Informationen nur implizit in den Ereignissequenzen vorhanden. Diese sollen explizit gemacht werden. Dies bedarf eines ausführlichen Preprocessings.

Zunächst findet, wie bei den MSPN, eine regelbasierte Aufteilung entlang der technischen Ressourcen und Filtrierung der nicht-relevanten Ereignisse statt. Darauf folgend werden Ereignisse von Signalen bestimmter Typen zu Ereignissen, die Aufschluss über Werkstückpositionen geben, kombiniert. Im Anschluss findet eine Zuordnung der Ereignisse zu Werkstückinstanzen statt; zunächst für jede technische Ressource und danach ressourcenübergreifend und anlagenweit. Diese Schritte sind im Folgenden erläutert.

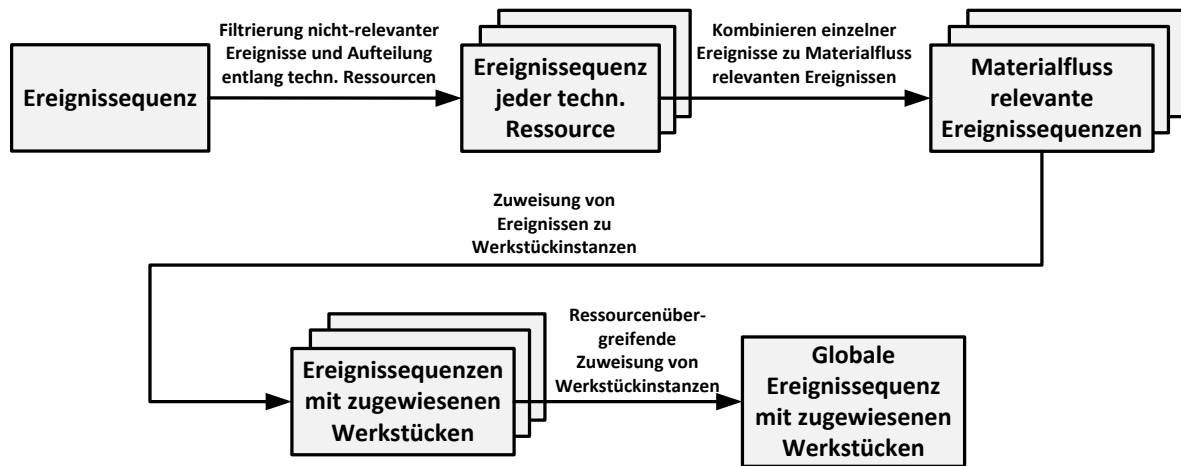


Abbildung 9-4: Preprocessing der Ereignissequenz zur Generierung eines MFPN

### 9.2.1 Aufteilung und Filtrierung der Ereignisse

Um eine Ereignissequenz zu erhalten, die lediglich für den Materialfluss relevantes Verhalten enthält, müssen zunächst die nicht-relevanten Signale aus der aufgenommenen Sequenz  $E_{obs}$  entfernt werden. Dies wird für jede technische Ressource vorgenommen. Als Resultat entsteht eine Menge an Ereignissequenzen, je eine für jede technische Ressource ( $E_{Mat}^r = (e_j^r(k))$ ), die ausschließlich für den Materialfluss relevantes Verhalten enthalten. Relevant sind dabei Ereignisse, die von Signalen stammen, die einen der Signaltypen  $S.XPosition$ ,  $S.YPosition$ ,  $S.WPIdentify$ ,  $S.WPDetect$  oder  $S.WPHold$  besitzen. Diese Signaltypen können Aufschluss über den Werkstücktypen oder die Werkstückposition geben, vgl. Abschnitt 7.3.

$$e_j^r(k) = \left\{ \left( \begin{array}{c} S_i^\alpha \\ k \cdot T_{OPC} \end{array} \right) : R(S_i) = R_r \wedge ST(S_i) \in MFRelevant \right\} \quad (9-9)$$

Mit:

$E_{Mat}^r$ : Technischer Ressource  $r$  zugewiesene beobachtete Ereignissequenz mit Materialfluss-relevanten Ereignissen

$R(S_i)$ : Im Informationsmodell Signal  $i$  zugewiesene technische Ressource

$ST(S_i)$ : Im Informationsmodell Signal  $i$  zugewiesener Signaltyp

$MFRelevant = \{S.XPosition, S.YPosition, S.WPIdentify, S.WPDetect, S.WPHold\}$ : Menge der für den Materialfluss relevanten Signaltypen

### 9.2.2 Kombinieren von einzelnen Ereignissen zu Materialfluss-relevanten Ereignissen

Während die Signaltypen  $S.WPIdentify$  und  $S.WPDetect$  explizite Informationen über die Werkstückposition bzw. den Werkstücktypen enthalten, ist dies bei den Signaltypen  $S.XPosition$ ,  $S.YPosition$  und  $S.WPHold$  nicht der Fall.  $S.WPHold$  wird Sensorsignalen zugewiesen, die angeben, dass eine technische Ressource ein Werkstück gegriffen hat (z.B. ein Kran).  $S.XPosition$  und  $S.YPosition$  geben Aufschluss über die Position der technischen Ressource. Hat die Ressource das Werkstück gegriffen, so ist die Werkstückposition gleichzusetzen mit der Position der technischen Ressource, was mithilfe einer geeigneten Verknüpfung der Signale ausgedrückt werden kann. Um die Signale zu verknüpfen, werden alle möglichen Kombinationen aus Signalen des Typen  $S.XPosition$  und  $S.YPosition$  mit dem Signal vom Typen  $S.WPHold$  UND-Verknüpft und daraus neue kombinierte Signale (siehe Listing 9-1).

```

1 Nehme Signal  $S_k$  vom Typen  $S.WPHold$ 
2 für alle Signale  $S_i$  vom Typen  $S.XPosition$ 
2   für alle Signale  $S_j$  vom Typen  $S.YPosition$ 
3     generiere neues Signal  $\tilde{S} = S_k \wedge S_i \wedge S_j$ 
4   ende
5 ende

```

**Listing 9-1: Pseudo-Code zur Generierung kombinierter Signale aus X- und Y-Position**

Existieren nur Signale von einem Typen,  $S.XPosition$  oder  $S.YPosition$ , so werden nur diese mit dem Signal vom Typen  $S.WPHold$  UND-verknüpft. Für jedes Ereignis in  $E_{Mat}^r$ , dass ein Signal der drei kombinierten Typen enthält, wird daraufhin geprüft, ob es einen Zustandswechsel eines zugehörigen kombinierten Signals zur Folge hat. Falls das Ergebnis der Prüfung positiv ausfällt, wird das Ereignis durch das entsprechende Ereignis des kombinierten Signals ersetzt. Falls nicht, so wird das Ereignis aus  $E_{Mat}^r$  entfernt. Folgendes Beispiel aus der in Abschnitt 11.1 diskutierten Fallstudie wird im Laufe dieses Kapitels als fortlaufendes Beispiel genutzt und soll zunächst o.g. Vorgehen verdeutlichen:

Der Kran der Fallstudienanlage bewegt sich zwischen drei horizontalen Endlagen. Diese werden mittels der Signale *Sensor\_CranOnMagazin* (Kran befindet sich am Magazin), *Sensor\_CranOnConveyor* (Kran befindet sich an Ausgangsrutsche) und *Sensor\_CranPositionStamp* (Kran befindet sich am Stempel) detektiert. Zudem existieren zwei vertikale Endlagen, die mittels der Signale *Sensor\_CranUp* (oben) und *Sensor\_CranDown* (unten) detektiert werden. Nimmt der Kran ein Werkstück auf, so wird dies mithilfe des Sensorsignals *Sensor\_CranSucked* erkannt. Den Signalen der horizontalen Endlagen wurde im Informationsmodell der Signaltyp  $S.XPosition$  zugewiesen und den Signalen der vertikalen Endlagen der Signaltyp  $S.YPosition$ . Dem Signal *Sensor\_CranSucked* wird der Signaltyp  $S.Hold$  zugewiesen. In Ruhelage befindet sich der Kran an oberer Position am Magazin. Beim Transport eines Werkstückes vom Magazin zur Ausgangsrutsche wurde eine Ereignissequenz aufgenommen (Tabelle 9-1):

**Tabelle 9-1: Beispielsequenz des Transports eines Werkstückes mit einem Kran**

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis
1	240	Sensor_CranUp	0
2	900	Sensor_CranDown	1
3	960	Sensor_CranDown	0
4	1141	Sensor_CranUp	1
5	2161	Sensor_CranUp	0
6	2821	Sensor_CranDown	1
7	2881	Sensor_CranSucked	1
8	3001	Sensor_CranDown	0
9	3181	Sensor_CranUp	1
10	3301	Sensor_CranOnMagazin	0
11	5161	Sensor_CranOnConveyor	1
12	5461	Sensor_CranUp	0
13	6061	Sensor_CranDown	1
14	6181	Sensor_CranDown	0
15	6181	Sensor_CranSucked	0

Mithilfe oben beschriebener Verknüpfung (Listing 9-1) der Signale werden nun sechs neue Signale generiert (zur vereinfachten Darstellung wurden bei allen Signalen die Präfixe „Sensor\_Cran“ entfernt):

- *Sucked&Down&OnMagazin*
- *Sucked&Up&OnMagazin*

- *Sucked&Down&OnConveyor*
- *Sucked&Up&OnConveyor*
- *Sucked&Down&OnStamp*
- *Sucked&Up&OnStamp*

Das „&“ im Signalnamen indiziert, dass die Signale UND-verknüpft wurden. Nun wird für obige Ereignissequenz geprüft, wann welches der neuen kombinierten Signale einen Zustandswechsel erfährt und, wenn dies der Fall ist, werden die einfachen Signale durch die kombinierten Signale ersetzt. Alle anderen Ereignisse der einfachen Signale werden entfernt. Dies führt zur in Tabelle 9-2 dargestellten Ereignissequenz. Die laufenden Nummern der Ereignisse in der ersten Spalte wurden dabei von der Ursprungssequenz übernommen, um den Bezug zu verdeutlichen.

Das erläuterte Verfahren zur Kombination von Ereignissen hilft, die Auflösung des später generierten MFPN zu erhöhen, ist jedoch nicht zwingend notwendig durchzuführen. Allerdings zeigt es, wie mithilfe der Signaltypen aus dem Informationsmodell allgemeine Regeln definiert und zur Generierung neuer Informationen angewandt werden können. Die Definition weiterer Regeln ist hier denkbar.

**Tabelle 9-2: Beispielsequenz nach Kombination der Signale**

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis
7	2881	Sucked&Down&OnMagazin	1
8	3001	Sucked&Down&OnMagazin	0
9	3181	Sucked&Up&OnMagazin	1
10	3301	Sucked&Up&OnMagazin	0
11	5161	Sucked&Up&OnConveyor	1
12	5461	Sucked&Up&OnConveyor	0
13	6061	Sucked&Down&OnConveyor	1
14/15	6181	Sucked&Down&OnConveyor	0

### 9.2.3 Lokale Zuweisung von Ereignissen zu Werkstückinstanzen

Um sich der Nebenläufigkeiten in der Ereignissequenz zu entledigen, werden die in der Sequenz enthaltenen einzelnen Transportvorgänge explizit gemacht. Hierzu wird ermittelt, welche Ereignisse demselben Transportvorgang entstammen. Jede Werkstückinstanz durchläuft das Fertigungssystem in einem spezifischen Transportvorgang. Daher dient der in diesem Abschnitt beschriebene Preprocessing-Schritt der Zuweisung von Ereignissen zu Werkstückinstanzen. Dies geschieht zunächst für jede technische Resource. Hierzu wird jedem Ereignis eine ID zugewiesen, die der Zuordnung von Ereignissen zu Werkstückinstanzen dient. Jedem Ereignis, das vom gleichen Werkstück ausgelöst wurde, wird entsprechend auch die gleiche ID zugewiesen. Dem im Folgenden erläuterten Verfahren liegen die in [FÜL14<sup>%</sup>] und [LFH+15<sup>#</sup>] beschriebenen Annahmen zugrunde<sup>33</sup>:

#### **Annahme 9-1:**

Der Transport eines Werkstücks zwischen zwei aufeinander folgenden detektierbaren Positionen verläuft zeitstabil. Das heißt, die Standardabweichung der Dauer zwischen dem Verlassen der Ausgangsposition und dem Erreichen der Zielposition ist sehr klein gegenüber der mittleren Transportdauer.

<sup>33</sup> Das in diesem Abschnitt aufgezeigte Verfahren entspricht von der Grundidee der in [FÜL14<sup>%</sup>; LFH+15<sup>#</sup>] bereits veröffentlichten Methode, wurde im Rahmen dieser Arbeit jedoch angepasst und ergänzt. Dies gilt insb. für die Anwendung der Clusteranalyse und den in Abschnitt 9.2.4 aufgezeigten Algorithmus.

**Annahme 9-2:**

Innerhalb einer technischen Ressource können Werkstücke sich nicht überholen, und Werkstücke können dieselbe Position ohne Verlassen der technischen Ressource nur einmal passieren, d.h. jedes Sensorsignal maximal einmal auslösen. Anlagenweit, d.h. ressourcenübergreifend, können Werkstücke jedoch Positionen mehrfach passieren.

Annahme 9-2 ist stets durch die Wahl einer geeigneten Granularität der Equipmenthierarchie im Informationsmodell erfüllbar. Die technischen Ressourcen auf unterster Ebene der Hierarchie müssen hierzu lediglich klein genug gewählt werden, z.B. einzelne Förderbänder. Annahme 9-1 dient der prinzipiellen Idee der Vorgehensweise: Ist die Transportdauer zwischen zwei detektierbaren Wegpunkten im System zeitstabil, so findet sich in der Ereignissequenz auch eine entsprechende Menge an Ereignispaaren, die stets die gleichen Ereignisse enthalten und (fast) gleiche Differenzen ihrer Zeitstempel aufweisen. Ziel des Algorithmus ist demnach für alle Ereignisse eine Gruppierung mit möglichst zeitstabilen Abständen gleicher Ereignisse zu finden. Das Finden dieser Gruppierung lässt sich als Optimierungsproblem über die Standardabweichungen der Zeitabstände zwischen den Ereignissen auffassen.

Die Schritte des Vorgehens werden anhand eines Beispiels, das der in Abschnitt 11.2 erläuterten Fallstudienanlage entstammt, erläutert. Im Beispielszenario wird das in Abbildung 9-5 schematisch dargestellte Teilsystem betrachtet. Es besteht aus mehreren Förderbändern und Drehtischen. Förderbänder sind mit einem oder zwei Sensoren zur Werkstückerkennung ausgestattet, Drehtische jeweils mit einem.

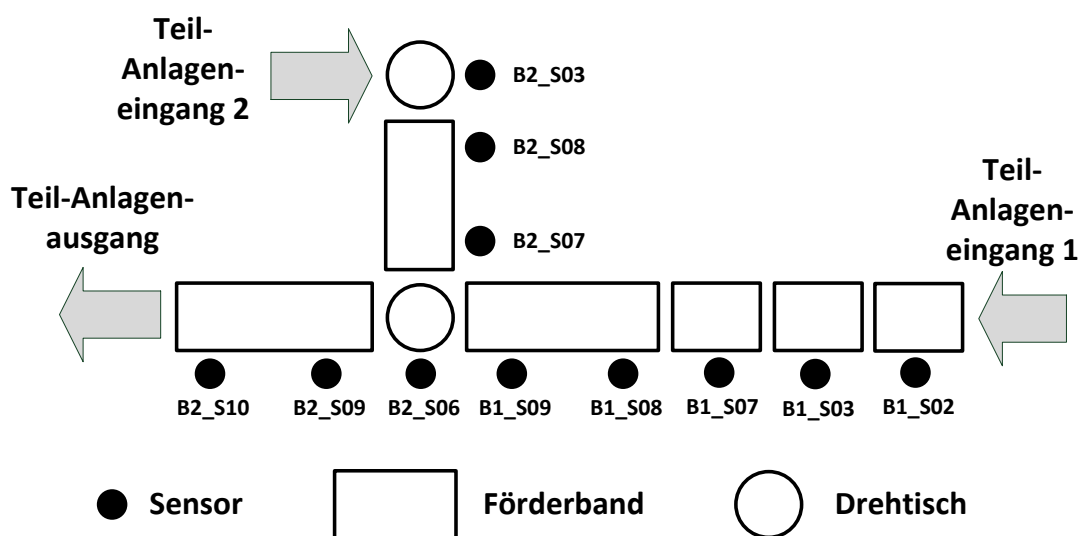


Abbildung 9-5: Schematische Darstellung des Fallstudien-Beispiels

Werkstücke können über zwei Eintrittspunkte in die Teilanlage geführt werden. Teil-Anlageneingang 1 entspricht dem Eintrittspunkt in die gesamte Anlage. Werkstücke, die den Anlagenabschnitt über den Teil-Anlagenausgang verlassen, werden u.U. über Teil-Anlageneingang 2 wieder in den Abschnitt eingebracht<sup>34</sup>. Ein Teil der Ereignissequenz, die sich aus dem Transport von drei Werkstücken ergibt, ist in

<sup>34</sup> Es sei zu beachten, dass Werkstücke, die eine Ressource verlassen und zu einem späteren Zeitpunkt wieder betreten, in diesem Schritt des Algorithmus zunächst als unterschiedliche Werkstückinstanzen betrachtet werden. Die Erkennung solcher Zusammenhänge ist Teil des im folgenden Abschnitt erläuterten Schritts des Algorithmus.

Tabelle 9-3 dargestellt<sup>35</sup>. Zwei Werkstücke wurden von Teil-Anlageneingang 1 zum Teil-Anlagenausgang transportiert und ein Werkstück von Teil-Anlageneingang 2 zum Teil-Anlagenausgang. Zwei der Transportvorgänge fanden dabei zur selben Zeit bzw. nebenläufig statt.

**Tabelle 9-3: Teil der Beispiel-Ereignissequenz**

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis
1	83974	'B1_S02'	1
2	87562	'B1_S02'	0
3	89278	'B1_S03'	1
4	90682	'B1_S03'	0
5	92242	'B1_S07'	1
6	93022	'B1_S07'	0
7	94895	'B1_S08'	1
8	95051	'B1_S08'	0
9	99107	'B1_S09'	1
10	99887	'B1_S09'	0
11	101759	'B2_S06'	1
12	102227	'B2_S06'	0
13	104255	'B2_S09'	1
14	104411	'B2_S09'	0
15	107375	'B2_S10'	1
16	107843	'B2_S10'	0
...	...	...	...
33	167279	'B2_S03'	1
34	168839	'B2_S03'	0
35	169775	'B1_S02'	1
36	170711	'B2_S08'	1
37	170867	'B2_S08'	0
38	172271	'B2_S07'	1
39	173363	'B1_S02'	0
40	174143	'B2_S07'	0
41	175079	'B1_S03'	1
42	175859	'B2_S06'	1
43	176327	'B1_S03'	0
44	177575	'B2_S06'	0
45	177887	'B1_S07'	1
46	178667	'B1_S07'	0
47	179603	'B2_S09'	1
48	179759	'B2_S09'	0
49	180383	'B1_S08'	1
50	180539	'B1_S08'	0
51	182723	'B2_S10'	1
52	183191	'B2_S10'	0
53	184751	'B1_S09'	1
54	185531	'B1_S09'	0
55	187403	'B2_S06'	1
56	187715	'B2_S06'	0
57	189743	'B2_S09'	1
58	190055	'B2_S09'	0
59	193019	'B2_S10'	1
60	193487	'B2_S10'	0
...	...	...	...

<sup>35</sup> Die vollständige Ereignissequenz mit 6 Werkstückdurchläufen ist in Anhang D zu finden. Die Ereignissequenz dort beinhaltet bereits die als Ergebnis des Algorithmus vergebenen IDs.

Im Folgenden sind die Schritte der ID-Zuweisung entsprechend der Nummerierung im Flussdiagramm des Verfahrens erläutert (Abbildung 9-6):

**1)** Im ersten Schritt der ID-Zuweisung wird jedem Vorkommnis des (kombinierten) Signals, das zum ersten Ereignis der Sequenz gehört, eine fortlaufende ID zugeteilt. Vorkommnis bedeutet hier eine steigende Flanke und die zugehörige folgende fallende Flanke desselben Signals. Da gem. Annahme 9-2 der Transport eines Werkstücks durch eine technische Ressource jedes Signal nur maximal einmal auslösen kann, handelt es sich auch bei jedem Vorkommnis um einen separaten Transportvorgang. In der gegebenen Ereignissequenz ist das erste Ereignis *B1\_S02* (Tabelle 9-3). Somit würden den Zeilen 1 und 2 die ID 1 zugewiesen und den Zeilen 35 und 39 eine andere ID (3, siehe Tabelle D-2 im Anhang).

**2)** Die folgenden Schritte 2 bis 10 werden solange ausgeführt, bis jedem Signal-Vorkommnis eine ID zugewiesen wurde. Hierbei werden in jedem Durchlauf die Vorkommnisse desjenigen Signals betrachtet, welches das erste Ereignis in der Sequenz ohne zugewiesene ID aufweist. Im Folgenden wird dieses Signal als „zu prüfendes Signal“ bezeichnet. Im obigen Beispiel ist dies zunächst das Signal *B1\_S03*.

**3)** Es werden die kürzesten Zeitabstände eines jeden Vorkommnisses des zu prüfenden Signals zu jedem Vorkommnis eines Signals mit bereits zugewiesener ID berechnet. Kürzester Zeitabstand bedeutet hier, dass der Zeitabstand zwischen steigender Flanke des zu prüfenden Signals und fallender Flanke des Signals mit bereits vergebener ID berechnet wird, wenn das Signalvorkommnis des zu prüfenden Signals nach dem Signal mit zugewiesener ID auftritt. Tritt das zu prüfende Signal vor dem Signal mit ID auf, wird der Abstand zwischen fallender Flanke des zu prüfenden Signals und steigender Flanke des Signals mit ID berechnet. Grund hierfür ist, dass der kürzeste Abstand weniger Störeinflüssen unterliegt. Zudem kann es möglich sein, dass der Abstand zwischen beiden steigenden Flanken variiert, zwischen fallender und steigender Flanke jedoch stabil ist. Dies ist z.B. der Fall, wenn ein Werkstück aufgrund variierender Wartezeit eine variable Dauer vor einem Sensor verweilt. Für obiges Beispiel sei dies an der Vergabe einer ID für die Signalvorkommnisse des Signals *B2\_S06* erläutert. Dieses wird hier als Beispiel herangezogen, da es in beiden möglichen Pfaden durch die Teilanlage ausgelöst wird. Nach mehreren Durchläufen des Algorithmus sind den Signalvorkommnissen der Signale *B1\_S02*, *B1\_S03*, *B1\_S07*, *B1\_S08* und *B1\_S09* bereits IDs (1-3) vergeben worden. In den Zeilen 1-10 wurden den Signalvorkommnissen jeweils die ID 1 vergeben. Damit ist *B2\_S06* in diesem Lauf erste Signal der Sequenz ohne zugewiesene ID und entsprechend das zu prüfende Signal. Im Beispiel befindet sich das erste Vorkommnis des Signals *B2\_S06* in den Zeilen 11 und 12 (Tabelle 9-3). Der kürzeste Abstand zu einem Vorkommnis eines Signals mit ID 1 ist die Differenz der Zeitstempel aus Zeile 11 und Zeile 10 und beträgt 1872 ms. Der kürzeste Abstand zu ID 3 ist die Differenz der Zeitstempel in den Zeilen 35 und 12 und beträgt 67548 ms. Für das Vorkommnis des Signals *B2\_S06* in den Zeilen 55 und 56 ergibt sich als kürzester Abstand zu ID 1 87516 ms und zu ID 3 1872 ms. Führt man diese Berechnung für alle Vorkommnisse des Signals *B2\_S06* zur Berechnung des zeitlichen Abstandes zum Signal *B1\_S03* durch<sup>36</sup>, so ergeben sich die in Tabelle 9-4 dargestellten Zeitdifferenzen (die oben erläuterten Zeitabstände sind grau hinterlegt).

---

<sup>36</sup> vgl. Tabelle D-2 im Anhang

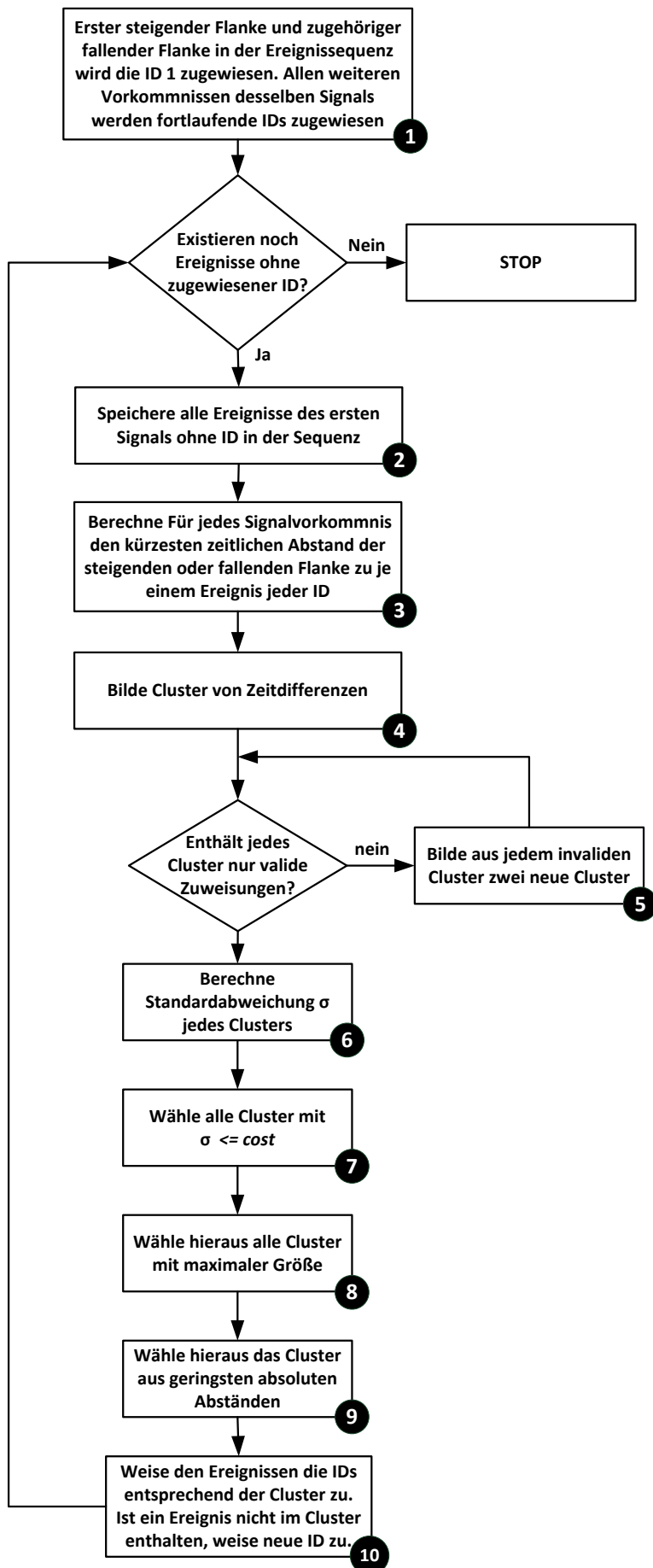


Abbildung 9-6: Flussdiagramm für die Zuweisung von IDs zu Ereignissen

Tabelle 9-4: Matrix aus Zeitdifferenzen (in ms) des Beispiels

Vorkommnis (Zeile)	ID		
	1	2	3
11/12	1872	21684	67548
27/28	41652	1872	27924
42/44	75972	36192	312
55/56	87516	47736	1872
67/69	133692	93912	48048
79/80	144144	104364	58500

4) Ziel ist es nun, eine Zuweisung der Signalvorkommnisse zu finden, die einer zeitstabilen Transportdauer entsprechen. Hierzu muss eine Menge an Zeitdifferenzen gefunden werden, die annähernd gleich sind. Eine Möglichkeit wäre es, alle Kombinationen von Zeitdifferenzen zu untersuchen [FÜL14<sup>6</sup>]. Dies erfordert jedoch einen sehr hohen Rechenaufwand, da die Anzahl an Zeitdifferenzen im ungünstigsten Fall im Quadrat der Anzahl an transportierten Werkstücken ansteigt. Unter Betrachtung der Zeitdifferenzen als Punkte auf einer Linie kann das Problem mithilfe der Clusteranalyse [BPW10] mit erheblicher Aufwandsersparnis gelöst werden [BRA15<sup>6</sup>]. Die Clusteranalyse dient der Klassifizierung bzw. Gruppierung von Wertepaaren bzw. Punkten. Beim hier verwendeten *K-means Clustering* wird eine Anzahl an Cluster ( $K$ ) vorgegeben, und der Algorithmus bestimmt die Cluster so, dass die Quadratsumme der euklidischen Abstände der Punkte innerhalb aller Cluster minimal ist ohne die Anzahl der Cluster zu verändern [BPW10]. Zur Verdeutlichung sind die in Tabelle 9-4 enthaltenen Zeitabstände in Abbildung 9-7 grafisch dargestellt. Die Farbe eines jeden Punktes in der Abbildung gibt an, welches Vorkommnis des zu prüfenden Signals herangezogen wurde. Die Abszisse beschreibt, zu welcher ID der Abstand bestimmt wurde, und die Ordinate gibt die entsprechende Zeitdifferenz an. Ein gebildetes Cluster entspricht einer möglichen Zuweisung der Vorkommnisse des zu prüfenden Signals gemäß der X-Werte der Punkte innerhalb des Clusters.

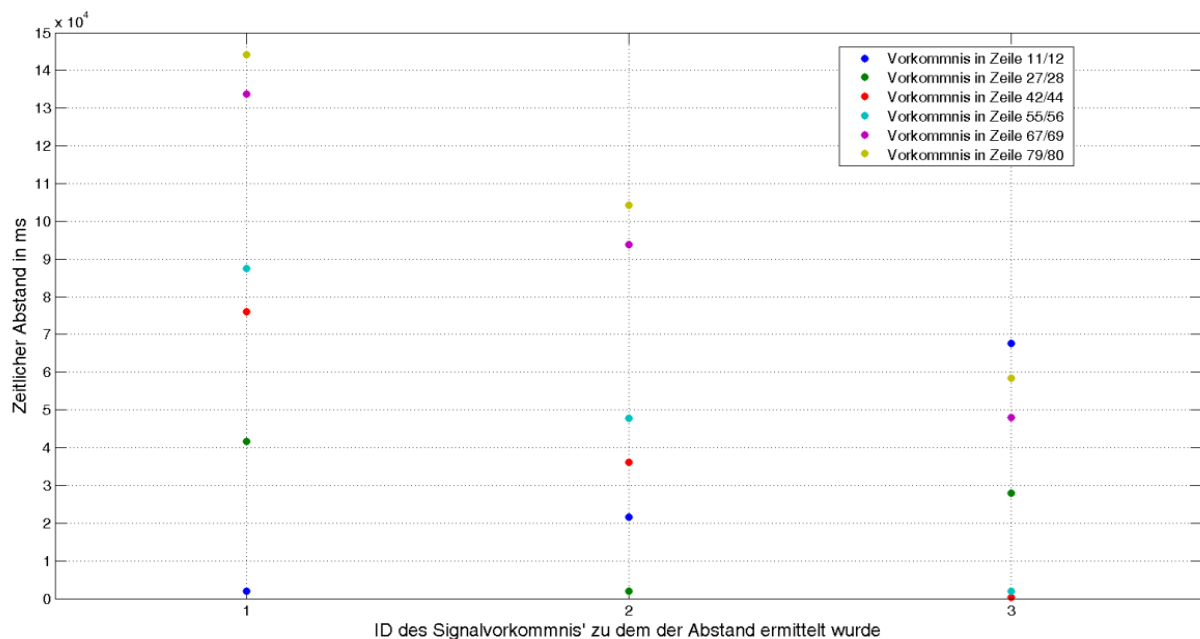


Abbildung 9-7: Grafische Darstellung der in Tabelle 9-4 gezeigten Zeitabstände

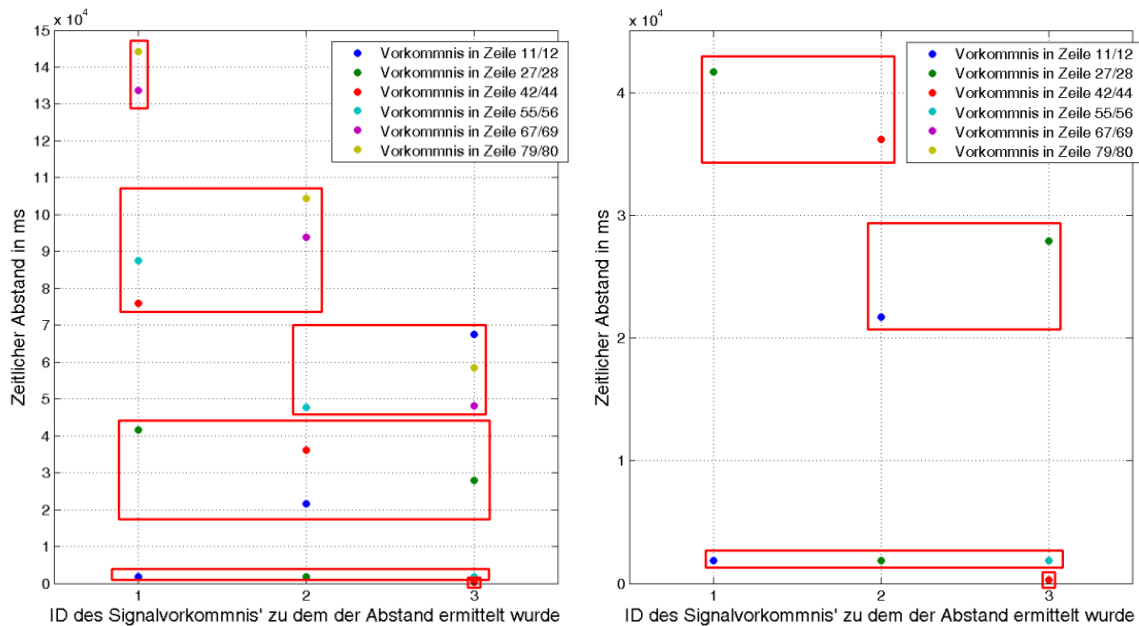
Da lediglich der Ordinaten-Wert für die Clusterung von Punkten mit ähnlichen Zeitdifferenzen relevant ist, entspricht der relevante euklidische Abstand dem vertikalen Abstand zwischen den Punkten. Als Anzahl zu bestimmender Cluster ( $K$ ) wird zunächst die Anzahl zu prüfender Signale gewählt (im Beispiel also sechs). Der Grund hierfür ist, dass ein Cluster nur als valide betrachtet werden kann, wenn jedes Vorkommnis des zu prüfenden Signals maximal einmal darin vorkommt. Kommt ein Vorkommnis des zu prüfenden Signals mehrfach in einem Cluster vor, so bedeute dies, dass das Vorkommnis mehreren IDs zuzuweisen wäre<sup>37</sup>, was offensichtlich nicht möglich ist und somit auf ein invalides Cluster hindeutet. Würden also aus allen Zeitdifferenzen weniger Cluster gebildet werden als Signalvorkommnisse, ist es unvermeidbar, dass mindestens ein invalides Cluster darunter ist. Um die Wahrscheinlichkeit der Bildung eines invaliden Clusters gering zu halten, muss  $K$  entsprechend groß gewählt werden. Eine zu große Anzahl an Clustern könnte jedoch dazu führen, dass Zeitdifferenzen, die sich eigentlich stark ähneln, in unterschiedliche Cluster gruppiert werden.

**5)** In diesem Schritt werden eventuell invalide gebildete Cluster identifiziert und iterativ zu mehreren Clustern verfeinert. Dies entspricht einer inkrementellen Anpassung der Anzahl an Clustern ohne Änderung bereits gefundener valider Cluster. Als invalide Cluster werden nicht nur diejenigen betrachtet, die mehr als ein Vorkommnis des zu prüfenden Signals enthalten, sondern auch diejenigen, in denen eine ID mehrfach vorkommt bzw. mehrere Signalvorkommnisse derselben ID zugewiesen werden sollen. Dies würde bedeuten, dass eine Werkstückinstanz das zu prüfende Signal mehrfach ausgelöst hat. Dies widerspricht jedoch Annahme 9-2. Zur Verdeutlichung sei nochmals auf das laufende Beispiel hingewiesen. Abbildung 9-8 zeigt auf der linken Seite die Gruppierung der Zeitabstände, die durch einmaliges Ausführen der Clusteranalyse mit  $K=6$  bestimmt wurde. Jedes identifizierte Cluster ist hier in einem roten Rechteck dargestellt. Die Anzahl der Cluster entspricht der Menge an Vorkommnissen des zu prüfenden Signals ( $B2\_S06$ ). Das Cluster mit den Zeitabständen zwischen 20000 ms und ca. 42000 ms entspricht dabei einem invaliden Cluster. Zum einen enthält es für mehrere IDs mehr als ein Signalvorkommnis. Dies ist daran zu erkennen, dass auf den vertikalen Linien innerhalb des Clusters mehr als ein Punkt enthalten ist. Eine diesem Cluster entsprechende Zuweisung würde bedeuten, dass zwei Vorkommnissen die ID 2 zugewiesen würde. Zum anderen sind Signalvorkommnisse mehrfach in dem Cluster enthalten. Dies ist wiederum daran zu erkennen, dass jeweils mehr als ein Punkt einer Farbe im Cluster enthalten ist. Zum Beispiel sind zwei grüne Punkte enthalten, je einer mit X-Wert (ID) 1 und 3. Eine entsprechende Zuweisung hieße, dass dem Signalvorkommnis sowohl die ID 1 als auch die ID 3 zugewiesen werden soll. Um die invaliden Cluster aufzulösen, werden diese weiter verfeinert, indem sie in je zwei neue Cluster dividiert werden. Hierzu wird für die Punkte innerhalb des invaliden Clusters die Clusteranalyse erneut mit  $K=2$  ausgeführt. Für obiges Beispiel eines invaliden Clusters ergeben sich dann die in Abbildung 9-8 auf der rechten Seite dargestellten zwei neuen Cluster (mit angepasster Skalierung der Ordinate). Dieser Schritt wird solange wiederholt, bis kein Cluster invalide Zuweisungen enthält. Im Beispiel zeigt sich, dass die beiden neu entstandenen Cluster nach der Division valide sind. Sie enthalten nur noch eine ID für jede Zuweisung und jedes Vorkommnis des zuzuweisenden Signals (resp. jede Farbe in der Darstellung) kommt auch nur noch einmal vor. Ergeben sich nach diesem Schritt Cluster mit nur noch einer Zuweisung (einem Punkt), so werden diese verworfen. Die darin enthaltenen Ereignisse werden in weiteren Durchläufen des Algorithmus wieder zur Analyse herangezogen. Dies geschieht ebenfalls mit Clustern, die in Schritt 4 bereits nur eine Zuweisung enthielten. Im obigen Beispiel gilt dies

---

<sup>37</sup> Es sei zu beachten, dass jedes Signalvorkommnis genau einmal für jeden X-Wert auftritt, da der Abstand des Vorkommnisses zur zugehörigen ID auch nur einmal berechnet werden kann. Kommt ein Signalvorkommnis also mehr als einmal in einem Cluster vor, so hat jeder zugehörige Punkt auch einen anderen X-Wert

bspw. für das Cluster, das dem Signalvorkommnis in den Zeilen 42 und 44 (rote Punkte) die ID 3 zuweist (zeitlicher Abstand 312 ms).



**Abbildung 9-8: Ursprünglich gebildete Cluster (linke Seite) und Cluster nach erster Korrektur (rechts)**

6) Alle Cluster weisen nun eine Menge von Zeitdifferenzen auf, die einer potentiell validen ID-Zuordnung entsprechen. Die zugrundeliegende Forderung einer zeitstabilen Transportdauer entspricht der Forderung nach einer möglichst kleinen Standardabweichung der Zeitdifferenzen innerhalb eines Clusters. Daher wird für alle ermittelten validen Cluster die Standardabweichung der in ihnen enthaltenen Zeitdifferenzen (Ordinatenwerte) ermittelt.

7) Da sich durch obige Vorgehensweise zufällig sehr kleine Cluster mit wenigen Punkten, die annähernd gleiche Zeitdifferenzen aufweisen, ergeben können, wird hier nicht direkt das Cluster gewählt, das die geringste Standardabweichung aufweist. Stattdessen werden alle Cluster, die eine Standardabweichung unterhalb eines definierten Schwellwertes besitzen, als Kandidaten für eine geeignete Zuweisung betrachtet. Dieses Vorgehen beruht auf der Annahme, dass die Transportdauer stets zeitstabil verläuft (siehe Annahme 9-1). Der Schwellwert entspricht einer erlaubten Standardabweichung in der Transportdauer von Werkstücken zwischen zwei detektierbaren Positionen. Er sollte daher leicht oberhalb einer zu erwartenden Ungenauigkeit im Transport zzgl. der zu erwartenden Signal-Latenzzeit liegen und muss ggf. empirisch oder heuristisch ermittelt werden. Für das hier behandelte Beispiel aus der Fallstudie war ein Schwellwert von 300 ms ausreichend, um ein valides Ergebnis zu erreichen. Dieser Wert wurde empirisch ermittelt.

8) Dieser Schritt dient dem Aussortieren valider Cluster mit einer Standardabweichung kleiner als der Schwellwert, die jedoch nur sehr wenige Punkte enthalten. Somit werden nur die Cluster weiter betrachtet, die eine möglichst hohe Anzahl an Punkten beinhalten. Dem liegt die Annahme zugrunde, dass Ereignisse, die öfter mit annähernd gleichen Zeitabständen beobachtet wurden, wahrscheinlicher einen durch den Materialfluss bedingten Zusammenhang aufweisen.

9) Sind nach vorigem Schritt noch mehr als ein Cluster als geeignete Kandidaten vorhanden, so wird jenes Cluster gewählt, das den kleinsten Mittelwert an Zeitabständen aufweist.

**10)** Entsprechend des nun ermittelten Clusters findet die Zuweisung der IDs statt. Für obiges Beispiel ist dies das Cluster mit Zeitdifferenzen von 1872 ms, vgl. Abbildung 9-8 rechts. Somit wird im Beispiel dem Signalvorkommnis des Signals  $B2\_S06$  in den Zeilen 11 und 12 der Ereignissequenz die ID 1 zugewiesen, dem Vorkommnis in den Zeilen 27 und 28 die ID 2 und den Signalvorkommnissen in den Zeilen 55 und 56 die ID 3<sup>38</sup>. Alle Signalvorkommnisse, denen in diesem Durchlauf keine ID zugewiesen wurden, werden in späteren Durchläufen des Algorithmus erneut berücksichtigt. Im Falle des Beispiels wird in späteren Durchläufen ein Zusammenhang zwischen dem Signal  $B2\_S07$  und den übrigen Vorkommnissen von  $B2\_S06$  erkannt.

#### 9.2.4 Verknüpfung lokaler Zuweisungen zu anlagenweiten globalen IDs

Nachdem für jede technische Ressource lokal die Zuweisung von Werkstückinstanzen zu Ereignissen in der entsprechenden Ereignissequenz  $E_{Mat}^r$  stattgefunden hat, müssen diese zu einer anlagenweiten Ereignissequenz  $E_{Mat}$  zusammengeführt werden. Dies macht es möglich, den Materialfluss ressourcenübergreifend widerzugeben und somit eine anlagenweite Abbildung und Analyse des Materialflusses vorzunehmen. Die zuvor lokal vergebenen IDs werden daher anlagenweiten globalen IDs zugeordnet, um schlussendlich die finale Zuweisung von Werkstückinstanzen zu Ereignissen vorzunehmen. Durch die vorangegangenen Aufwände der Zuweisung einer lokalen ID ist bereits für einen Großteil an Ereignissen ein Zusammenhang ermittelt worden. Dies reduziert den Aufwand zur Vergabe von globalen IDs. Allerdings kann anlagenweit nicht mehr davon ausgegangen werden, dass Annahme 9-2 zutrifft, da dies zu einer zu starken Restriktion des Ansatzes führen würde. Somit muss der Algorithmus in der Lage sein, wiederkehrende Positionen im Transport einer Werkstückinstanz zu erkennen.

Da Ressourcen-intern bereits die Zusammenhänge zwischen den Ereignissen ermittelt wurden, werden lediglich das erste und letzte Ereignis einer jeden internen ID betrachtet. Diese werden im Folgenden als *Eintrittsereignisse* (erstes Ereignis einer ID in einer technischen Ressource) und *Austrittsereignisse* (letztes Ereignis einer ID in einer technischen Ressource) bezeichnet, da sie den Eintritt eines Werkstückes in eine technische Ressource bzw. den Austritt eines Werkstückes aus einer technischen Ressource indizieren. Aus allen Ereignissequenzen  $E_{Mat}^r$  werden für jede interne ID die Eintritts- und Austrittsereignisse in eine zur Zuweisung der globalen IDs generierten Ereignissequenz  $E_{Mat}^{glob}$  geschrieben. Tabelle 9-5 zeigt einen Teil der so entstandenen Ereignissequenz für ein Beispiel, das der Fallstudie aus Abschnitt 11.1 entnommen wurde<sup>39</sup>. Neben den üblichen Informationen (Zeitstempel, Signalname und Ereignis) enthält die Sequenz eine zur Ressourcenzuweisung genutzte Nummer sowie die zuvor ermittelte interne ID. Die Benchmarkanlage besteht in diesem Beispiel aus drei Anlagenteilen<sup>40</sup>: Dem bereits vorgestellten Kran (Ressourcennummer 1), einem Eingangsmagazin (Ressourcennummer 2) sowie einem Stempel (Ressourcennummer 3).

Das erste Eintrittsereignis des Kranes mit ID 1 findet sich bspw. in Zeile 2 und das zugehörige Austrittsereignis in Zeile 5<sup>41</sup>. Aufgabe des Algorithmus ist es, kausale Zusammenhänge zwischen Austrittsereignissen einer Ressource und Eintrittsereignissen der anderen Ressourcen zu identifizieren. Dies wird, wie bei der Zuweisung lokaler IDs, mithilfe der Ermittlung von zeitstabilen Abständen der Ereignisse vorgenom-

<sup>38</sup> vgl. Tabelle D-2 im Anhang

<sup>39</sup> Die vollständige Sequenz befindet sich in Anhang D.

<sup>40</sup> Die in Abschnitt 11.1 beschriebene Ausgangsrutsche ist in diesem Szenario rein mechanisch und enthält keine Sensor- oder Aktorsignale und wird daher vom Algorithmus nicht weiter betrachtet.

<sup>41</sup> Vgl. auch die vollständige Tabelle D-3 im Anhang. Diese beinhaltet zudem bereits das Ergebnis der Vergabe globaler IDs

men. Das Vorgehen ist in einem Flussdiagramm dargestellt (Abbildung 9-9) und wird im Folgenden erläutert:

**Tabelle 9-5: Beispiel-Ereignissequenz  $E_{Mat}^{glob}$  zur Zuweisung globaler IDs**

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis	Ressource	Interne ID
1	1921	WorkpieceReady	1	2	1
2	2881	CranSucked&CranOnMagazin&CranDown	1	1	1
3	2881	WorkpieceReady	0	2	1
4	5701	MagazinCapacitiveSensor	1	2	2
5	6181	CranSucked&CranOnConveyor&CranDown	0	1	1
6	9181	CranSucked&CranOnMagazin&CranDown	1	1	2
7	9241	WorkpieceReady	0	2	2
8	12241	WorkpieceReady	1	2	3
9	14162	StampSliderFilled	1	3	1
10	14402	CranSucked&CranPositionStamp&CranDown	0	1	2
11	17042	CranSucked&CranPositionStamp&CranDown	1	1	7
12	17222	StampSliderFilled	0	3	1
13	20401	CranSucked&CranOnConveyor&CranDown	0	1	7

**1)** Zunächst wird das erste Austrittsereignis in  $E_{Mat}^{glob}$  ohne globale ID bestimmt. Dies ist im initialen Durchlauf die erste fallende Flanke in der Sequenz. In obigem Beispiel (Tabelle 9-5) ist dies im ersten Durchlauf das Signal *WorkpieceReady* in Zeile 3.

**2)** Allen gleichen Austrittsereignissen (im Beispiel in den Zeilen 3 und 7) werden fortlaufende IDs zugewiesen. Da ein Wiedereintritt einer Werkstückinstanz in die Ressource nicht ausgeschlossen werden kann, ist es möglich, dass hier eine fehlerhafte Zuweisung stattfindet. In diesem Fall würden mehrere Austrittssignale zur selben Werkstückinstanz gehören. Entsprechend müssten sie auch derselben ID zugewiesen werden. Dies wird später im Algorithmus geprüft und ggf. korrigiert, siehe Schritt 10.

**3)** Das aktuelle Austrittsereignis muss nun auf Zeitstabilität zu allen Eintrittsereignissen geprüft werden, die nicht von derselben Ressource stammen. Daher werden die folgenden Schritte 4 und 5 für jeden Ressourcenkandidaten durchgeführt. Im Beispiel wird das Austrittsereignis *WorkpieceReady* des Magazins (Ressource 2) geprüft. Somit muss es gegen die Eintrittsereignisse des Kranes (Ressource 1) und des Stempels (Ressource 3) geprüft werden.

**4)** Nun wird eine eventuell vorhandene Zeitstabilität zwischen dem aktuellen Austrittsereignis und allen Eintrittsereignissen der zu prüfenden Ressource ermittelt. Dies geschieht, indem die Schritte 3 bis 9 der Zuweisung lokaler IDs (Abbildung 9-6) für das aktuelle Austrittsereignis und alle Eintrittsereignisse der Ressource ausgeführt werden. Im Beispiel besitzt der Stempel die steigende Flanke des Signals *StampSliderFilled* als Eintrittsereignis. Die Clusteranalyse ergibt, dass keine Zeitstabilität mit einer Standardabweichung kleiner der hier vorgegebenen Schwelle von 300 ms zum Austrittsereignis des Signals *WorkpieceReady* existiert. Der Kran besitzt hier zwei Eintrittsereignisse, die steigenden Flanken der Signale *CranSucked&CranOnMagazin&CranDown* und *CranSucked&CranPositionStamp&CranDown*. Für ersteres kann ein Cluster mit einer Zeitstabilität entsprechend einer absoluten Standardabweichung von 20,33 ms ermittelt werden.

**5)** Das durch Ausführung der Clusteranalyse am besten geeignete Cluster der Ressource wird gespeichert, um später gegen die Cluster der anderen Ressourcen verglichen zu werden.

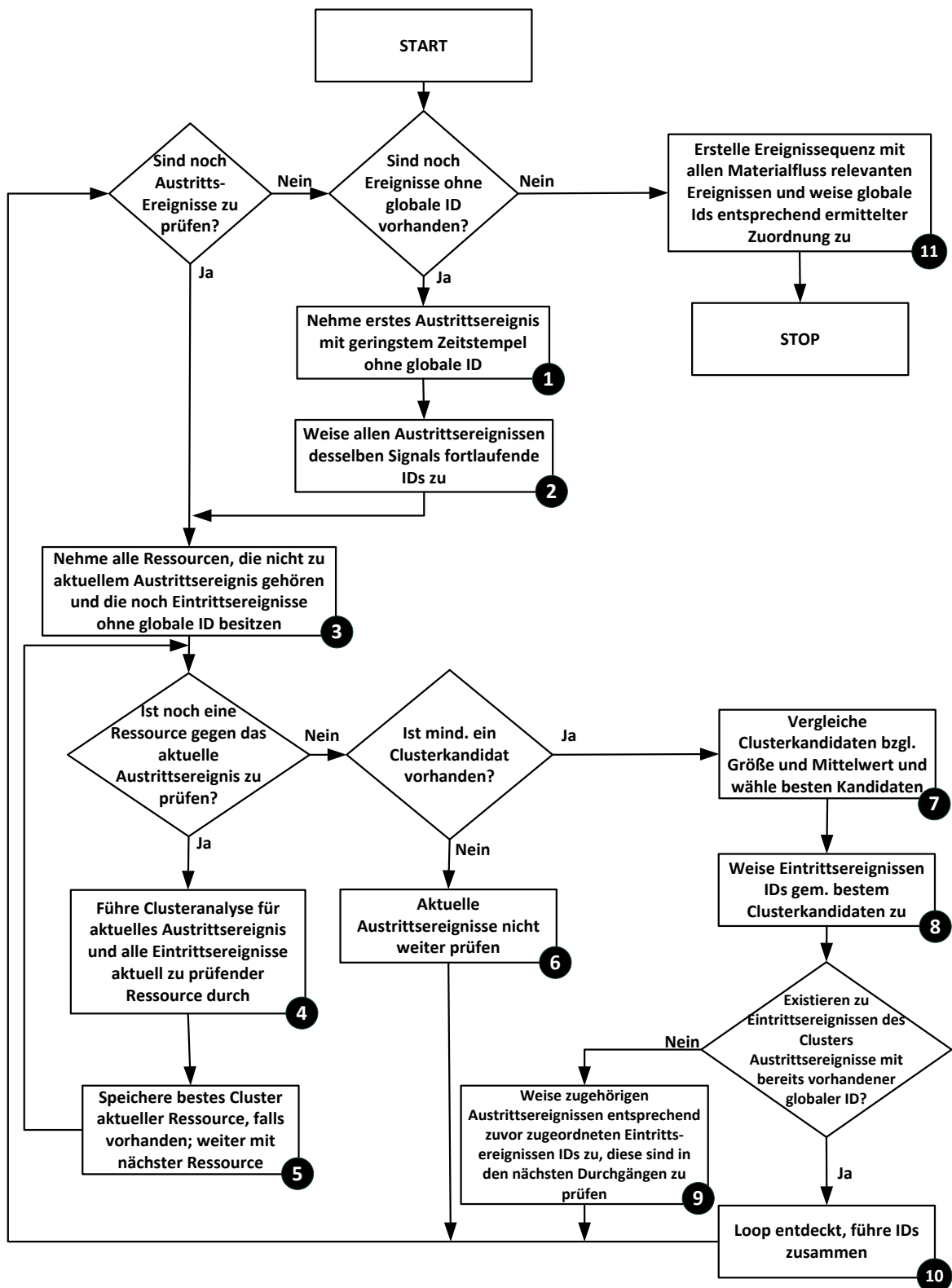


Abbildung 9-9: Flussdiagramm zur Bestimmung globaler IDs

6) Ist für keine Ressource ein geeignetes Cluster gefunden, so werden die aktuellen Austrittsereignisse nicht weiter auf Zeitstabilität zu Eintrittsereignissen geprüft. Wenn keine Zeitstabilität zu Eintrittsereignissen besteht, heißt dies, dass kein Eintrittsereignis auf das Austrittsereignis folgt und die Austrittsereignisse den Austritt aus der gesamten Anlage detektieren.

- 7) Sind nun Clusterkandidaten von mehreren Ressourcen vorhanden, so wird, analog zur Auswahl des besten Clusterkandidaten für die Vergabe lokaler IDs, der Clusterkandidat mit maximaler Größe und geringstem Mittelwert der Zeitabstände gewählt. Im Beispiel existiert nur ein Clusterkandidat, der als bester Kandidat gespeichert wird. Dies ist oben genanntes Cluster, das einer Zuweisung des Austrittssignals *WorkpieceReady* zum Eintrittssignal *CranSucked&CranOnMagazin&CranDown* entspricht.
- 8) Entsprechend des ermittelten besten Clusters werden den Eintrittsereignissen die globalen IDs zugewiesen. Im Beispiel erhält Zeile 2 die globale ID, die zuvor Zeile 3 zugewiesen wurde und Zeile 7 erhält dieselbe ID wie Zeile 6. Sind geprüfte Austrittsereignisse nicht im Cluster enthalten, werden diese im nächsten Durchgang nochmals geprüft.
- 9) Sind den Austrittsereignissen, die zu den gerade zugewiesenen Eintrittsereignissen gehören (d.h. die dieselbe Ressourcenzuordnung und interne ID besitzen, resp. die zur selben Werkstückinstanz gehören) noch keine IDs zugewiesen, so werden den Austrittsereignissen die IDs entsprechend der zugehörigen Eintrittsereignisse zugewiesen. Im Beispiel erhält das Ereignis in Zeile 5 dieselbe ID wie das Ereignis in Zeile 2, gleiches gilt für die Zeilen 6 und 10. Die so neu zugeordneten Austrittsereignisse werden in den nächsten Durchgängen des Algorithmus geprüft.
- 10) Sollte ein Eintrittsereignis bereits ein Austrittsereignis mit vergebener ID besitzen (durch fortlaufende Vergabe in Schritt 2), so liegt das daran, dass eine Werkstückinstanz eine Ressource mehrfach durchläuft. Entsprechend müssen die globalen IDs zu einer zusammengeführt werden.
- 11) Sind alle Ereignisse in  $E_{Mat}^{glob}$  mit globalen IDs versehen, werden allen weiteren Materialflussrelevanten Ereignissen entsprechend globale IDs zugewiesen und in einer anlagenweiten Ereignissequenz  $E_{Mat}$  zusammengetragen. Jedem Ereignis wird dabei genau die globale ID vergeben, die dem Ereignis aus  $E_{Mat}^{glob}$  mit selber Ressourcenzuordnung und interner ID vergeben wurde.

Tabelle 9-6 zeigt obigen Ausschnitt von  $E_{Mat}^{glob}$  nach Vergabe der globalen ID<sup>42</sup>. Es ist zu erkennen, dass die Werkstückinstanz mit globaler ID 2 (grau hinterlegt) den Kran (Ressource 2) zweimal durchläuft. Im ersten Durchlauf transportiert der Kran das Werkstück vom Magazin zum Stempel und im zweiten Durchlauf vom Stempel zur Ausgangsrutsche. Der Algorithmus konnte somit erfolgreich ein wiederholtes Durchlaufen einer Werkstückinstanz durch eine Ressource erkennen.

**Tabelle 9-6: Beispiel-Ereignissequenz  $E_{Mat}^{glob}$  nach Vergabe globaler IDs**

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis	Ressource	Interne ID	Globale ID
1	1921	WorkpieceReady	1	2	1	1
2	2881	CranSucked&CranOnMagazin&CranDown	1	1	1	1
3	2881	WorkpieceReady	0	2	1	1
4	5701	MagazinCapacitiveSensor	1	2	2	2
5	6181	CranSucked&CranOnConveyor&CranDown	0	1	1	1
6	9181	CranSucked&CranOnMagazin&CranDown	1	1	2	2
7	9241	WorkpieceReady	0	2	2	2
8	12241	WorkpieceReady	1	2	3	3
9	14162	StampSliderFilled	1	3	1	2
10	14402	CranSucked&CranPositionStamp&CranDown	0	1	2	2
11	17042	CranSucked&CranPositionStamp&CranDown	1	1	7	2
12	17222	StampSliderFilled	0	3	1	2
13	20401	CranSucked&CranOnConveyor&CranDown	0	1	7	2

<sup>42</sup> die vollständige Sequenz sowie die aus Schritt 11 resultierende Gesamtsequenz  $E_{Mat}$  befinden sich in Anhang D

### 9.3 Generieren des Petri-Netzes

Die durch obiges Verfahren gewonnene Ereignissequenz  $E_{Mat}$  wird nun genutzt, um das MFPN zu generieren. Die Generierung besteht dabei aus 5 Schritten: (1) Generieren eines Petri-Netzes für jede Werkstückinstanz, (2) Zusammenfassen von Stellen mit demselben Prä-Ereignis, (3) Zusammenfassen von Transitionen mit selben Vor- und Nachstellen, (4) Auflösen von Mehrdeutigkeiten und (5) Generieren der Zeit-, Mengen und Identifikationsannotationen. Diese Schritte werden in den folgenden Absätzen erläutert. Da das laufende Beispiel zur Veranschaulichung nur bedingt geeignet ist, wird zunächst auf ein fiktives Beispiel zurückgegriffen und das aus der Fallstudie entnommene Beispiel erst in Schritt 5 wieder aufgenommen. Im fiktiven Beispiel können Werkstücke zwei Transportrouten durch ein System nehmen. Die erste Route ist beschrieben durch die Signalfolge  $S1 \rightarrow S2 \rightarrow S3 \rightarrow S2$  und die zweite Route durch die Signalfolge  $S1 \rightarrow S2 \rightarrow S4 \rightarrow S3 \rightarrow S5$ <sup>43</sup>. Diese wurden als Ereignissequenzen aufgenommen und entsprechend unterschiedlichen IDs zugewiesen.

#### 1) Generieren eines Petri-Netzes je Werkstückinstanz

Zunächst wird die Ereignissequenz  $E_{Mat}$  nach den globalen IDs in Subsequenzen  $E_{Mat}^j$  separiert, wobei  $j$  ein Index über alle IDs ist. Für jede so entstandene Subsequenz, die jeweils die zurückgelegte Transportroute der zugehörigen Werkstückinstanz beinhaltet, wird ein Petri-Netz erzeugt. Hierfür wird für die erste steigende Flanke jeder Sequenz  $E_{Mat}^j$  eine Transition  $\mathcal{T}_t$  und eine zugehörige Nachstelle  $\mathcal{S}_s \bullet (\mathcal{T}_t)$  generiert. Die Transition wird mit dem entsprechenden Signalnamen  $\tilde{S}(\mathcal{T}_t)$  annotiert. Gemäß der Reihenfolge in  $E_{Mat}^j$  wird für jede folgende steigende Flanke eine weitere Transition mit Nachstelle an die letzte Stelle des Netzes angehängt. Ereignisse, die von Signalen des Typen  $S.WPIdentify$  stammen, sind hiervon ausgenommen. Diese werden in Schritt 5 zur Generierung der Zeit-, Mengen- und Identifikationsannotationen benötigt. Es entsteht eine Transitionen-Stellen-Kette deren annotierte Signale der Reihenfolge an steigenden Flanken in  $E_{Mat}^j$  entspricht. Für das fiktive Beispiel ergeben sich somit zwei Petri-Netze. Jedes der Netze gibt eine in der Ereignissequenz enthaltene Transportroute wieder. Die annotierten Signale an den Transitionen in Markenflussrichtung entsprechen den beobachteten Signalfolgen  $S1 \rightarrow S2 \rightarrow S3 \rightarrow S2$  sowie  $S1 \rightarrow S2 \rightarrow S4 \rightarrow S3 \rightarrow S5$  (Abbildung 9-10).

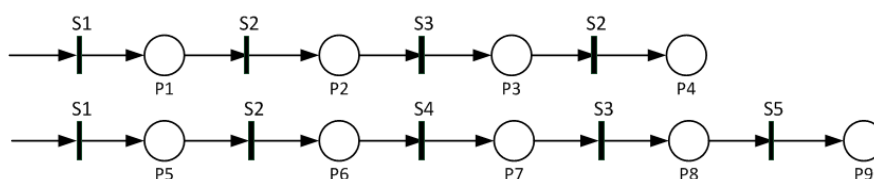


Abbildung 9-10: Resultierende Petri-Netze nach Ausführung von Schritt 1

#### 2) Zusammenfassen von Stellen mit selben Prä-Ereignis

Die zuvor entstandenen einzelnen Petri-Netze je Werkstückinstanz (im Folgenden *Instanz-Netz*) werden nun zusammengefasst, sofern sie Ähnlichkeiten aufweisen. Hierzu werden alle Stellen, die Vortransitionen mit dem gleichen annotierten Signal besitzen, zu einer Stelle zusammengefasst. Stellen werden dabei nicht mit Stellen desselben Instanz-Netzes zusammengefasst. Sollte ein Signal mehr als einmal in einem Instanz-Netz annotiert sein, so wird diejenige Stelle zusammengefasst, die mehr Gleichheit mit der Stelle des anderen Instanz-Netzes in Form von Signalen, die an weiteren Vortransitionen annotiert

<sup>43</sup> Signalfolge heißt hier, dass zu jedem Signal steigende und fallende Flanken beobachtet wurden, hier jedoch der Einfachheit halber nur die Reihenfolge der steigenden Flanken dargestellt ist

sind, aufweist. Hierfür werden die Signalspuren verglichen, die sich durch die Annotationen der Transitionen entgegen der Markenflussrichtung ergeben. In Abbildung 9-10 bspw. haben im oberen Netz die Stellen  $P4$  und  $P2$  je eine Vortransition, die mit  $S2$  annotiert ist. Im unteren Netz gilt dies für  $P6$ . Es muss also entschieden werden, welche der beiden Stellen,  $P2$  oder  $P4$ , mit  $P6$  zusammengeführt wird. Hierfür werden weitere Vortransitionen betrachtet. Im Falle von  $P4$  folgt  $S2$  auf  $S3$  und im Falle von  $P2$  folgt  $S2$  auf  $S1$ . Da  $S2$  auch für  $P6$  auf  $S1$  folgt, werden hier  $P2$  und  $P6$  zusammengeführt. Das resultierende Netz des Beispiels ist im oberen Teil von Abbildung 9-11 dargestellt.

### 3) Zusammenfassen von Transitionen mit denselben Vor- und Nachstellen

Besitzen mehrere Transitionen dieselbe Vor- und Nachstelle, so handelt es sich auch um den gleichen Teil derselben Transportroute. Daher werden in diesem Schritt alle Transitionen, die exakt die gleichen Vor- und Nachstellen besitzen, zusammengefasst. Für das fiktive Beispiel gilt dies für die Transition zwischen  $P1$  und  $P2$  (Abbildung 9-11).

### 4) Auflösen von Mehrdeutigkeiten

Nach Durchführung der vorangegangenen Schritte können Mehrdeutigkeiten in Form von Stellen mit mehreren eingehenden und mehreren ausgehenden Kanten auftreten. Diese müssen aufgelöst werden, um der Restriktion (9-6) gerecht zu werden. Daher werden für jede dieser Stellen alle möglichen Signalfolgen bestimmt, die sich durch Kombination der Annotationen an den Vor- und Nachtransitionen ergeben. Da Stellen nur zusammengeführt werden, wenn sie dasselbe Prä-Ereignis besitzen (siehe Schritt 2), müssen auch hier weitere Vortransitionen berücksichtigt werden, um die Signalfolgen unterscheiden zu können. Im Beispiel zeigt  $P4$  eine Mehrdeutigkeit auf (siehe Abbildung 9-11, unten). Mögliche Signalfolgen der Vor- und Nachtransitionen von  $P4$  sind  $S3 \rightarrow S2$  und  $S3 \rightarrow S5$ . Unter Betrachtung weiterer Vortransitionen können vier mögliche Signalfolgen unterschieden werden:  $S2 \rightarrow S3 \rightarrow S2$ ,  $S2 \rightarrow S3 \rightarrow S5$ ,  $S4 \rightarrow S3 \rightarrow S2$  und  $S4 \rightarrow S3 \rightarrow S5$ .

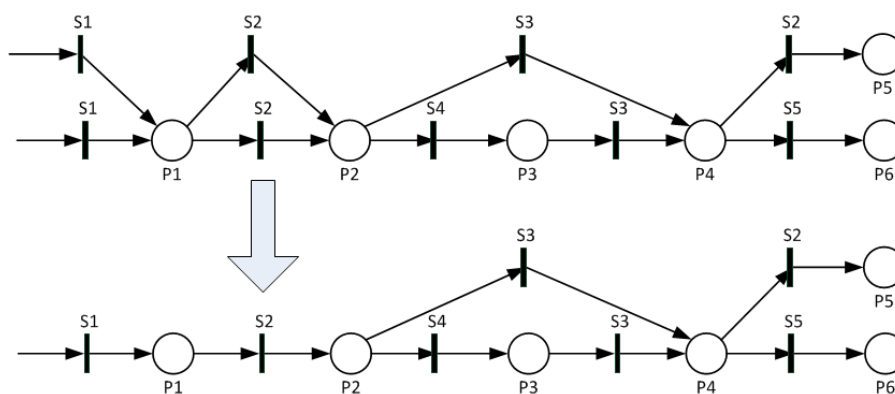


Abbildung 9-11: MFPN des fiktiven Beispiels nach Schritt 2 (oben) und Schritt 3 (unten)

Zunächst wird verglichen, wie viele der möglichen Signalfolgen tatsächlich in den Subsequenzen  $E_{Mat}^j$  auftreten. Im Beispiel sind dies die beiden Folgen  $S2 \rightarrow S3 \rightarrow S2$  und  $S4 \rightarrow S3 \rightarrow S5$ . Die Stelle, die die Mehrdeutigkeit aufweist, und all ihre eingehenden und ausgehenden Kanten werden nun kopiert. Es werden so viele Duplikate erstellt, bis die Stelle so oft vorhanden ist, wie tatsächlich aufgekommene Signalfolgen existieren. Im Beispiel kommen zwei Signalfolgen tatsächlich vor, somit muss  $P4$  einmal kopiert werden. Das Ergebnis dieses Zwischenschrittes ist im oberen Teil von Abbildung 9-12 grafisch dargestellt.

Jeder der nun mehrfach vorkommenden mehrdeutigen Stelle wird je eine der tatsächlich beobachteten Signalfolgen zugewiesen. Im Anschluss werden diejenigen Kanten entfernt, die nicht in der Signalfolge

resultieren, die der Stelle zugewiesen ist. In obigem Beispiel wird  $P4$  die Signalfolge  $S4 \rightarrow S3 \rightarrow S5$  und  $P7$  die Signalfolge  $S2 \rightarrow S3 \rightarrow S2$  zugewiesen. Entsprechend werden die Kanten, die nicht zu diesen Signalfolgen führen aus dem Netz entfernt, wie in Abbildung 9-12 unten dargestellt.

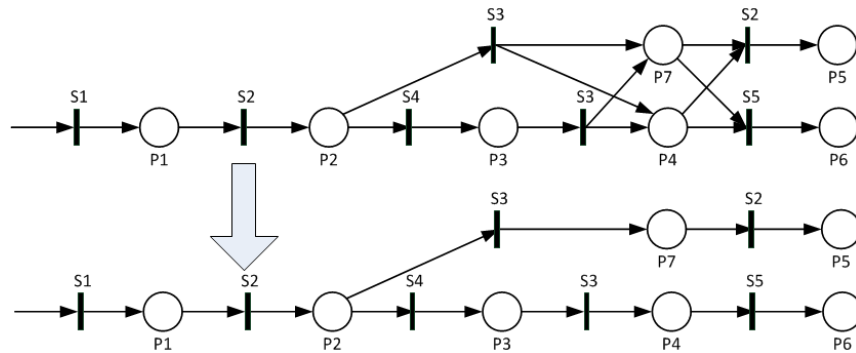


Abbildung 9-12: Schritt 4 der MFPN-Generierung am fiktiven Beispiel

### 5) Generieren der Zeit-, Mengen und Identifikationsannotationen

Der letzte Schritt der MFPN-Generierung annotiert den Vektor aus 6-Tupeln  $\langle d_{min}, d_{max}, \mu, \sigma, n, \mathbf{ident} \rangle$  an Transitionen, Stellen und Kanten gem. der Definition in Abschnitt 9.1. Hierzu wird zunächst für jedes Ereignis aller Sequenzen  $E_{Mat}^j$  ermittelt, welcher Transition es zuzuordnen ist. Da einige Signale an mehreren Transitionen annotiert sein können, müssen hier, ähnlich wie in den Schritten 2 und 4, die vorangehenden und nachfolgenden Ereignisse in  $E_{Mat}^j$  mit den Vor- und Nachtransitionen der mit selbigem Signal annotierten Transition verglichen werden, um eine eindeutige Zuordnung treffen zu können.

Zur Bestimmung der **ident**-Vektoren wird für jedes Signal der Anlage, dem der Typ  $S.WPIdentify$  im Informationsmodell zugewiesen wurde, ein Element für die **ident**-Vektoren vorgesehen und mit dem Wert 0 initialisiert. Für jede Subsequenz  $E_{Mat}^j$  wird ein zugehöriger **ident**-Vektor identifiziert. Hierzu wird geprüft, welche Signale mit Typ  $S.WPIdentify$  in  $E_{Mat}^j$  vorkommen und der zugehörige **ident**-Vektor wird an den entsprechenden Stellen mit einer 1 beschrieben. Für das ebenfalls in den Abschnitten 9.2.1, 9.2.2 und 9.2.4 behandelte Fallstudienbeispiel existiert im dargestellten Szenario ein Signal vom Typen  $S.WPIdentify$ . Dieses heißt *Sensor\_MagazinCapacitiveSensor*. Somit entspricht das einzige Element der **ident**-Vektoren den Elementarzuständen dieses Signals:

$$\mathbf{ident} = (\mathcal{Z}(\text{Sensor\_MagazinCapacitiveSensor})), \quad (9-10)$$

Mit:

$$\mathcal{Z}(x) \in \{0,1\}: \text{Elementarzustand des Signals } x$$

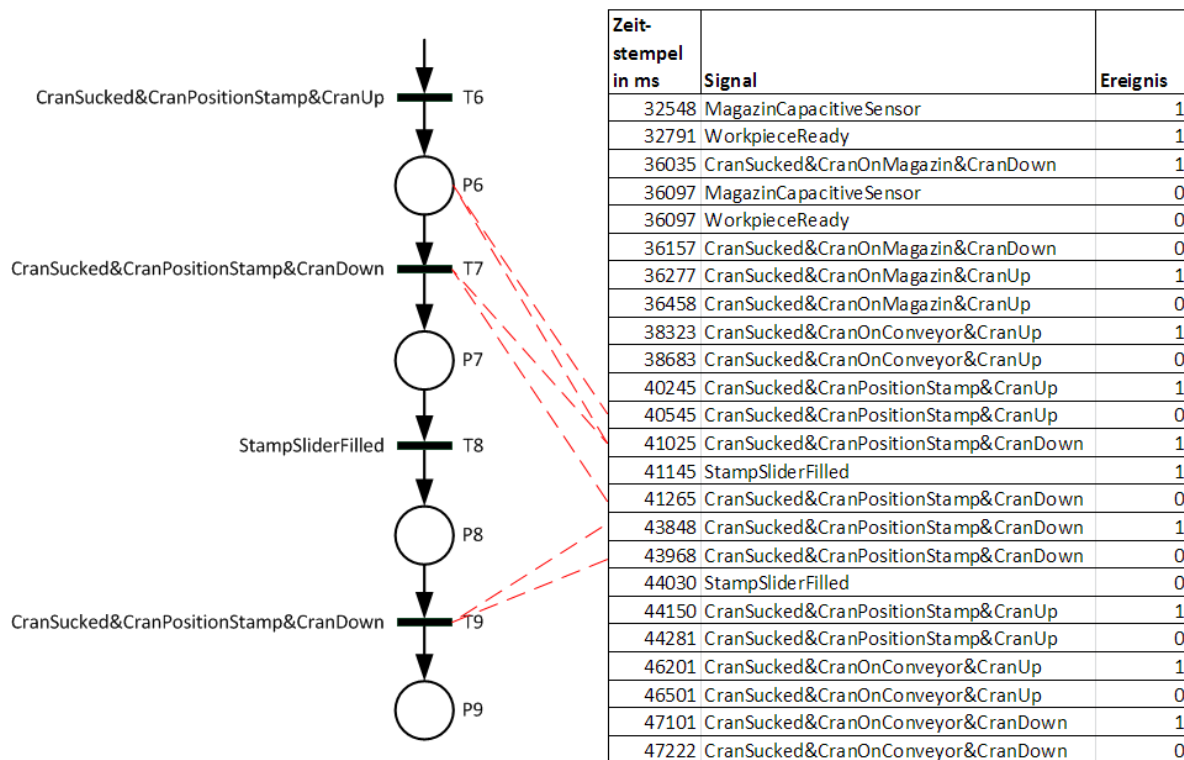
Für jedes weitere Signal des Typen  $S.WPIdentify$  würde der Vektor um ein Element, das den Elementarzustand des Signals repräsentiert, ergänzt werden. Im Preprocessing des Beispiels wurden sechs Werkstückinstanzen identifiziert. Zwei davon weisen ein Vorkommen des Signals *Sensor\_MagazinCapacitiveSensor* auf, die restlichen vier nicht. Daher wird für die beiden Ereignissequenzen mit Vorkommen des Signals *Sensor\_MagazinCapacitiveSensor* je ein **ident**-Vektor (1) angelegt und für die restlichen vier der **ident**-Vektor (0). Der Grund, warum Identifikationssignale in Subsequenzen nicht auftreten, kann zum einen sein, dass die zugehörigen Werkstücke den entsprechenden Sensor in ihrer Transportroute nie passiert haben. Zum anderen ist es möglich, dass sie den Sensor nicht ausgelöst haben. Ersterer Fall führt dazu, dass Werkstücke gleichen Typs unterschiedliche **ident**-Vektoren auf ihrer Transportroute generieren. Um dennoch Werkstücke gleichen Typs über die Identifikationssignale zu erkennen, ist es möglich eine manuell angelegte Matching-Tabelle zu verwenden, in der hinterlegt ist,

welche Sensorsignale Aufschluss über gleiche Eigenschaften eines Werkstückes geben. Alternativ könnte dies auch in Form von Eigenschaften der Signale im Informationsmodell hinterlegt werden. Ähnlich könnte bspw. vorgegangen werden, wenn Werkstücke über Bar- oder RFID<sup>44</sup>-Codes identifiziert werden und bestimmte Bereiche der Codes gleiche Werkstücktypen identifizieren.

Zur Berechnung der Zeitinformationen (Transportdauern) wird für jede Subsequenz  $E_{Mat}^j$  die Dauer eines jeden Signalvorkommnisses berechnet. Hierzu wird für jedes Vorkommnis die Differenz der Zeitstempel von fallender und steigender Flanke des Signals berechnet. Genauso wird die Zeitdifferenz zwischen steigender Flanke eines jeden Signals und der fallenden Flanke des vorangehenden Signals ermittelt, um Stellen und Kanten zu annotieren. Abbildung 9-13 verdeutlicht dies anhand des Fallbeispiels. Auf der linken Seite ist ein Ausschnitt des generierten Petri-Netzes dargestellt und auf der rechten Seite eine der Subsequenzen, aus denen das Netz generiert wurde. Die Transitionen  $T7$  und  $T9$  sind jeweils mit dem Signal *CranSucked&CranPositionStamp&CranDown* annotiert. Die nächst vorangehende Transition zu  $T7$  ist  $T6$ , die mit dem Signal *CranSucked&CranPositionStamp&CranUp* annotiert ist. Die nächst vordere Transition zu  $T9$  ist  $T8$ , welche wiederum mit dem Signal *StampSliderFilled* annotiert ist. Somit wird jedes Signalvorkommnis von *CranSucked&CranPositionStamp&CranDown*, dem ein Vorkommnis von *CranSucked&CranPositionStamp&CranUp* vorangeht, der Transition  $T7$  zugewiesen. Diejenigen Signalvorkommnisse von *CranSucked&CranPositionStamp&CranDown*, denen *StampSliderFilled* vorangeht, werden Transition  $T9$  zugewiesen. Dies ist in Abbildung 9-13 anhand der gestrichelten Linien angedeutet. Der Stelle  $P6$  müssen die Zeitdifferenzen zwischen den Vorkommnissen der Signale *CranSucked&CranPositionStamp&CranUp* (fallende Flanke) und *CranSucked&CranPositionStamp&CranDown* (steigende Flanke) zugewiesen werden, sofern diese in der Subsequenz direkt aufeinander folgen. Auch dies ist mittels einer gestrichelten Linie angedeutet.

Sind für alle Transitionen, Stellen und (ggf.) Kanten (im Folgenden allgemein als Elemente bezeichnet) die Transportzeiten für jede Werkstückinstanz bestimmt, so werden die Zeiten zur Ermittlung der Zeit- und Mengenannotationen genutzt. Hierzu werden alle Zeiten, die demselben Element zugewiesen wurden und aus Sequenzen  $E_{Mat}^j$  mit gleichem ermitteltem *ident*-Vektor stammen, zusammengefasst. Die Anzahl zusammenfassender Zeiten mit gleichem *ident*-Vektor entspricht der Anzahl an Werkstücken des durch *ident* beschriebenen Werkstücktypen, die den durch das Element repräsentierten Transportabschnitt passiert haben. Diese Anzahl entspricht daher dem zu annotierenden Wert  $n$ . Aus den Zeiten werden dann der Minimalwert  $d_{min}$ , der Maximalwert  $d_{max}$ , der Mittelwert  $\mu$  und die Standardabweichung  $\sigma$  berechnet und gemeinsam mit *ident* als 6-Tupel am Element annotiert.

<sup>44</sup> Radio-frequency identification



**Abbildung 9-13: Teil des aus dem Fallbeispiel entstandenen Petri-Netzes und Zuweisung von Transitionen zu Signalvorkommnissen in der Ereignissequenz**

Abbildung 9-14 zeigt einen Abschnitt des generierten MFPN mit allen Annotationen. In den geschweiften Klammern sind die 6-Tupel dargestellt, die an den nächst gelegenen Elementen annotiert sind. Die Werte sind exakt in der Reihenfolge  $\{d_{min}, d_{max}, \mu, \sigma, n, \mathit{ident}\}$  angegeben und alle Zeitwerte sind in Millisekunden dargestellt und auf 1ms gerundet. Zu erkennen ist, dass die Transitionen T1 bis T4 sowie die Stellen P1 bis P3 mit je zwei 6-Tupeln annotiert sind. Je eines enthält dabei den *ident*-Vektor [0] und je eines den *ident*-Vektor [1], womit jedes 6-Tupel für den so identifizierten Werkstücktypen zu interpretieren ist. Es lässt sich anhand der Werte für *n* erkennen, dass vier Werkstücke mit *ident*-Vektor [0] beobachtet wurden und zwei mit *ident*-Vektor [1], was auch dem tatsächlichen Werkstückmengen entspricht. Die Stelle P4 enthält zwei ausgehende Kanten. Daher ist an ihr kein 6-Tupel annotiert, sondern jeweils an den beiden ausgehenden Kanten. Zu erkennen ist, dass Werkstücke mit *ident*-Vektor [1] die Route über die detektierte Position *CranSucked&CranPositionStamp&CranUp* nehmen und Werkstücke mit *ident*-Vektor [0] die Route über Position *CranSucked&CranOnConveyor&CranDown*. Die negativen Zeitwerte an P1 sind dadurch zu erklären, dass hier eine Überlappung der Signale *WorkpieceReady* und *CranSucked&CranOnMagazin&CranDown* stattfand. Das heißt, die steigenden Flanken des Signals *CranSucked&CranOnMagazin&CranDown* haben einen früheren Zeitstempel als die fallenden Flanken des Signals *WorkpieceReady* in den entsprechenden Ereignissequenzen. Damit die Transportdauer mittels Addition der Zeitannotationen berechnet werden kann und auch entsprechend korrekt ist, müssen diese Werte negativ sein, was sich auch aus der beschriebenen Berechnung ergibt.

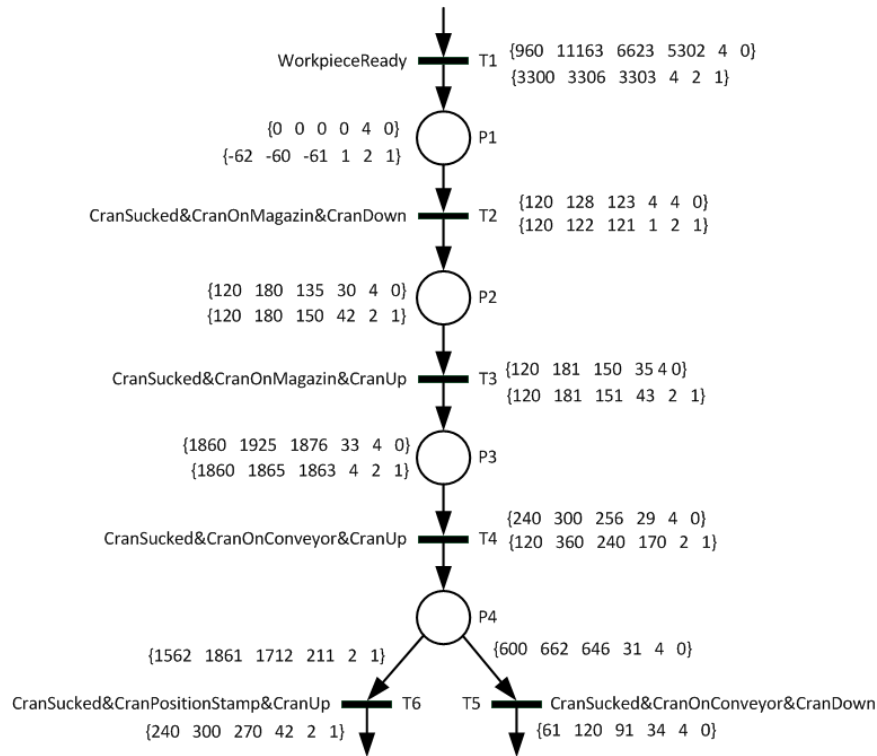


Abbildung 9-14: Teil des für das Fallbeispiel generierten MFPN inkl. Annotationen

### 9.4 Anomalieerkennung

Zur Erkennung von Anomalien im Materialfluss wird das Verhalten des MFPN mit dem aktuell beobachtbaren Verhalten des Materialflusses verglichen. Hierzu wird dem MFPN jedes zum Zeitpunkt  $k \cdot T_{OPC}$  auftretende für den Materialfluss relevante Ereignis  $e_j(k)$  zugeführt. Um für den Materialfluss relevant zu sein, muss das Ereignis einem Signal entstammen, das einen für den Materialfluss relevanten Signaltypen besitzt:

$$e_j(k) \in \left\{ \left( \begin{matrix} S_i^\alpha \\ k \cdot T_{OPC} \end{matrix} \right) : ST(S_i) \in MFRelevant \right\} \tag{9-11}$$

Mit:

$ST(S_i)$ : Im Informationsmodell Signal  $i$  zugewiesener Signaltyp

$MFRelevant = \{S.XPosition, S.YPosition, S.WPIdentify, S.WPDetect, S.WPHold\}$ : Menge der für den Materialfluss relevanten Signaltypen

Entsprechend jedem eingehenden Ereignisses wird, sofern möglich, die Markierung des MFPN  $m_k$  neu berechnet. Ist das Ereignis Teil eines im MFPN kombinierten Ereignisses, so wird stets geprüft, ob durch das aufgetretene Ereignis auch ein kombiniertes Ereignis ausgelöst wurde (vgl. Abschnitt 9.2.2).

Tritt ein Ereignis auf, nicht auf oder zu einem Zeitpunkt auf, der dem Verhalten des MFPN im aktuellen Zustand widerspricht, so wird dies als eine Anomalie gewertet. Aus dieser Aussage können Anomalien abgeleitet werden, die einem MFPN widersprüchliches Verhalten beschreiben. Folgend werden fünf Anomalien vorgestellt, die auf eine evolutionäre Änderung hinweisen:

1. Eine steigende Flanke  $s_i^\uparrow$  eines Signals tritt auf, das an keiner aktivierten Transition im MFPN annotiert ist:

$$\nexists \mathcal{T}_t \in MFPN : \mathcal{T}_t \text{ aktiviert} \wedge (s_i \subseteq \tilde{S}(\mathcal{T}_t)) \quad (9-12)$$

In diesem Fall hieÙe dies, dass eine neue Transportroute beobachtet wurde, da ein Werkstück eine Signalfolge auslöst, die zuvor nicht beobachtet wurde, z.B. aufgrund neu eingesetzten Materialfluss-Equipments oder eines neu implementierten Routings.

2. Eine steigende Flanke  $s_i^\uparrow$  eines Signals, das an einer aktivierten Transition im MFPN annotiert ist, tritt zum Zeitpunkt  $k \cdot T_{OPC}$  auf; allerdings ist der zeitliche Abstand zur fallenden Flanke des Signals, das an der zuvor gefeuerten Transition annotiert ist, geringer als die minimale Transportdauer annotiert an der Vorstelle bzw. der Kante über die die Marke in die Stelle geflossen ist oder der Kante zwischen der Vorstelle und der aktivierten Transition dessen steigende Flanke beobachtet wurde<sup>45</sup>:

$$\begin{aligned} \exists \mathcal{T}_t \in MFPN, e_j(k-n) = \left( \begin{array}{c} s_l^\downarrow \\ (k-n) \cdot T_{OPC} \end{array} \right) : \\ \mathcal{T}_t \text{ aktiviert} \wedge (s_i \subseteq \tilde{S}(\mathcal{T}_t)) \wedge (s_l \subseteq \tilde{S}(\bullet(\bullet\mathcal{T}_t))) \wedge (n \cdot T_{OPC} > d_{min}(\bullet\mathcal{T}_t)) \end{aligned} \quad (9-13)$$

Mit:

$d_{min}(\bullet\mathcal{T}_t)$ : Minimale Transportdauer annotiert an der Vorstelle von  $\mathcal{T}_t$  bzw. der Kante zwischen der Vorstelle und  $\mathcal{T}_t$  oder der Kante zwischen  $\mathcal{T}_t$  und der Stelle über die die Marke in die Vorstelle geflossen ist

Mit anderen Worten verweilt eine Marke kürzer in einer Stelle, als an ihr oder der Kante, über die die Marke abfließt oder zugeflossen ist, annotiert ist. Diese Anomalie weist darauf hin, dass ein Transportvorgang zwischen zwei detektierbaren Positionen schneller vonstattengeht als durch das Modell vorgesehen, z.B. aufgrund einer Drehzahlerhöhung von Motoren im Materialflussequipment.

3. Eine Marke verweilt länger in einer Stelle als die Maximaltransportdauer die an ihr oder allen ausgehenden oder eingehenden Kanten der Stelle annotiert ist:

$$\begin{aligned} \exists e_{j-m}(k-n) = \left( \begin{array}{c} s_l^\downarrow \\ (k-n) \cdot T_{OPC} \end{array} \right) \wedge \nexists e_j(k) = \left( \begin{array}{c} s_i^\uparrow \\ k \cdot T_{OPC} \end{array} \right) : \\ (s_l \subseteq \tilde{S}(\mathcal{T}_t)) \wedge (s_i \subseteq \tilde{S}((\mathcal{T}_t \bullet) \bullet)) \wedge (n \cdot T_{OPC} < d_{max}(\mathcal{T}_t \bullet)) \end{aligned} \quad (9-14)$$

Tritt diese Anomalie auf, so ist ein bestimmter Transportweg entweder gar nicht mehr vorhanden (z.B. entferntes/defektes Materialflussequipment) oder die Transportdauer hat sich erhöht (z.B. aufgrund von Verschleiß).

4. Eine fallende Flanke  $e_j(k) = \left( \begin{array}{c} s_l^\downarrow \\ (k) \cdot T_{OPC} \end{array} \right), s_i \subseteq \tilde{S}(\mathcal{T}_t)$  tritt auf, aber die Zeitdifferenz zur zugehörigen steigenden Flanke  $e_{j-m}(k-n) = \left( \begin{array}{c} s_i^\uparrow \\ (k-n) \cdot T_{OPC} \end{array} \right)$  ist kleiner als die an der zugehörigen aktivierten Transition annotierte minimale Aufenthaltsdauer  $d_{min}(\mathcal{T}_t)$ :

<sup>45</sup> Es sei zu beachten, dass je nach Anzahl der eingehenden und ausgehenden Kanten entweder eine Kante oder eine Stelle annotiert ist, jedoch nie beides (Abschnitt 9.3). Vereinfacht wird die zum aktuellen Markenfluss zugehörige Annotation mit  $d_{min}(\bullet\mathcal{T}_t)$  notiert

$$\exists e_j(k) = \begin{pmatrix} s_i^\downarrow \\ (k \cdot T_{OPC}) \end{pmatrix}, e_{j-m}(k-n) = \begin{pmatrix} s_i^\uparrow \\ (k-n) \cdot T_{OPC} \end{pmatrix}, s_i \subseteq \tilde{S}(\mathcal{J}_t) : \quad (9-15)$$

$$(n \cdot T_{OPC} < d_{min}(\mathcal{J}_t))$$

Anhand dieser Anomalie kann eine verkürzte Verweildauer an einer detektierbaren Position erkannt werden. Dies könnte z.B. Folge einer verkürzten Bearbeitungszeit sein.

5. Eine fallende Flanke  $e_j(k) = \begin{pmatrix} s_i^\downarrow \\ (k) \cdot T_{OPC} \end{pmatrix}, s_i \subseteq \tilde{S}(\mathcal{J}_t)$  tritt auf, aber die Zeitdifferenz zur zugehörigen steigenden Flanke  $e_{j-m}(k-n) = \begin{pmatrix} s_i^\uparrow \\ (k-n) \cdot T_{OPC} \end{pmatrix}$  ist größer als die an der zugehörigen Transition annotierte minimale Aufenthaltsdauer  $d_{max}(\mathcal{J}_t)$ :

$$\exists e_j(k) = \begin{pmatrix} s_i^\downarrow \\ (k) \cdot T_{OPC} \end{pmatrix}, e_{j-m}(k-n) = \begin{pmatrix} s_i^\uparrow \\ (k-n) \cdot T_{OPC} \end{pmatrix}, s_i \subseteq \tilde{S}(\mathcal{J}_t) : \quad (9-16)$$

$$(n \cdot T_{OPC} > d_{max}(\mathcal{J}_t))$$

Anhand dieser Anomalie kann eine verlängerte Verweildauer an einer detektierbaren Position erkannt werden, z.B. aufgrund verlängerter Bearbeitungszeiten oder eines angepassten Routings, dass höhere Verweildauern an bestimmten Positionen zur Folge hat.

## 10 Eigenschaftsbestimmung

Abschnitt 3.3 hat Kennzahlen und Metriken zu Qualitätsmerkmalen aufgezeigt, die zur Bewertung von Fertigungssystemen herangezogen werden können und aus der Literatur und Praxis bekannt sind. Hieraus wurden Messelemente bestimmt, die zur Ermittlung der Merkmalsausprägungen benötigt werden<sup>46</sup>. In diesem Kapitel soll nun die Möglichkeit zur Ermittlung oder Schätzung der Messelemente aus den generierten Modellen diskutiert werden. Da nicht alle Messelemente aus den Modellen bestimmt werden können (z.B. sind keine Informationen über Maschinenausfälle in den Modellen enthalten), liegt das Hauptaugenmerk auf Zeiten und zeitbezogenen Größen, Topologie- und Strukturgrößen sowie den produzierten Mengen. Aus den weiteren Merkmalsklassen werden, wenn möglich, einzelne Größen bestimmt. Diese Kennzahlen können bereits zur Bestimmung einer Vielzahl von relevanten Eigenschaften beitragen oder zumindest eine Tendenz der Änderung der Eigenschaften angeben. Zur Bestimmung der Messelemente werden sowohl Struktur und Verhalten der generierten MSPN und MFPN herangezogen als auch die im Informationsmodell gegebenen Semantiken zur Interpretation der Modelle. Des Weiteren werden zur Analyse auch die zu den Modellen zugehörigen Ereignissequenzen  $E^r$  und  $E_{Mat}$  sowie die beobachtete Rohsequenz  $E_{obs}$  herangezogen.

### 10.1 Ermittlung der Positionen zur Werkstückbearbeitung

Für einige Kenngrößen ist es notwendig zu bestimmen, an welcher Stelle im Materialfluss eine Werkstückbearbeitung stattfindet. Daher soll in diesem Abschnitt zunächst aufgezeigt werden, wie mithilfe der Modelle und Ereignissequenzen dieser Zusammenhang ermittelt werden kann.

Das Verhalten der Ressourcen ist durch die zugehörigen MSPN beschrieben. Jedes MSPN, das mindestens ein Signal vom Typen *A.WPModify* enthält, beschreibt also eine Maschine, die Operationen an Werkstücken durchführt. Enthält die Maschine selbst ein oder mehrere Signale mit einem Materialfluss relevanten Typen, so kommen diese Signale bzw. Ereignisse der Signale sowohl im zugehörigen MSPN als auch im MFPN vor. Die Position der Bearbeitung ist dann durch die Transition(en) im MFPN gegeben, an denen diese Signale annotiert sind. Ist dies nicht der Fall, so ist es notwendig eine Invariante aus 1-Zuständen eines Signals in  $E_{Mat}$  und dem 1-Zustand des Signals vom Typ *A.WPModify* zu ermitteln. Es muss also eine Position gefunden werden, an der sich stets ein Werkstück befindet, wenn der Aktor zur Werkstückbearbeitung eingeschaltet wurde.

#### Beispiel:

Für die in Abschnitt 11.2 beschriebene Fallstudie wurde ein MFPN mit 26 Transitionen, also 26 detektierten Positionen generiert. Die Anlage enthält mehrere Werkzeugmaschinen, die Werkstücke bearbeiten. Die Werkstücke befinden sich während der Bearbeitung auf Förderbändern, die wiederum vor den Werkzeugmaschinen stehen. Abbildung 10-1 zeigt dies anhand eines Beispiels aus der Teilanlage „Slave 3“. Werkstücke werden bearbeitet, wenn sie sich auf Förderband 3.3 befinden, was mithilfe des Sensors B3\_S09 erkannt wird. Da sowohl für jedes Förderband als auch für die Werkzeugmaschine je ein eigenes MSPN gelernt wurde, geben die Modelle keinen Zusammenhang zwischen der Position „B3\_S09“ und der Bearbeitung durch diese Maschine wieder. Um diesen Zusammenhang zu ermitteln, werden alle Ereignissequenzen, die Signale mit Typ *A.WPModify* beinhalten, herangezogen. Dies sind exakt diejeni-

---

<sup>46</sup> Tabelle C-1 im Anhang

gen Ereignissequenzen, die zum Lernen der MSPNs für die Werkzeugmaschinen genutzt wurden<sup>47</sup>. Für jede Sequenz wird zunächst ermittelt, wann die Signale vom Typen *A.WPModify* eine steigende Flanke aufweisen.

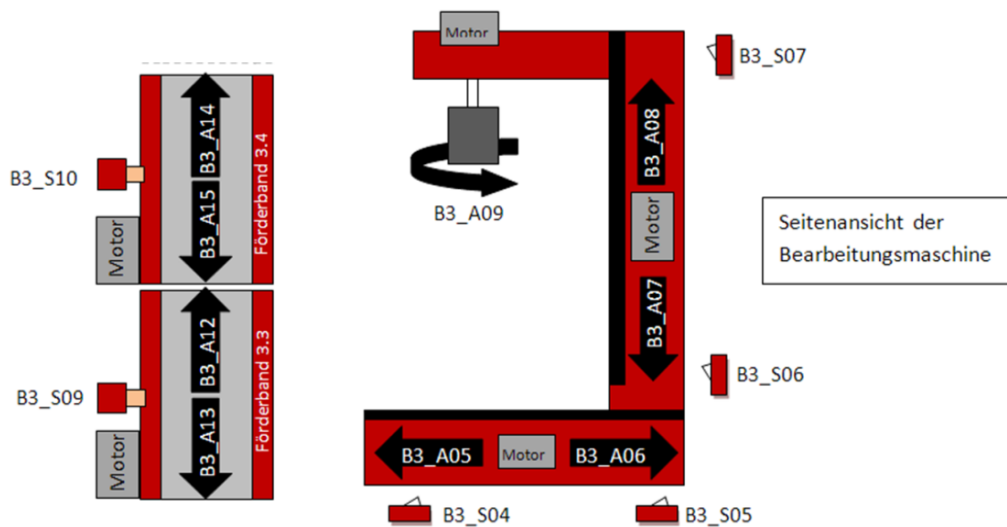


Abbildung 10-1: Beispiel aus Teilanlage "Slave 3" der Fallstudie

Für die Beispielsequenz in Tabelle D-5 zeigt das Signal B3\_A09 vier steigende Flanken mit den Zeitstempeln 68219 ms, 204875 ms, 425459 ms und 601327 ms auf. Anschließend wird geprüft, welche Positionen zu diesen Zeitpunkten detektiert wurden, also welche Signale in  $E_{Mat}$  zu diesen Zeitpunkten einen 1-Zustand aufwiesen<sup>48</sup>. Ein Vergleich ergibt, dass ausschließlich das Signal B3\_S09 immer einen 1-Zustand aufweist, wenn der Aktor B3\_A09 aktiviert ist. Somit kann geschlossen werden, dass die Bearbeitung mittels B3\_A09 an der Stelle B3\_S09 stattfindet. Eine Eingrenzung der Position kann auch mithilfe der Equipmenthierarchie im Informationsmodell vorgenommen werden, wenn diese mehr als eine Hierarchieebene enthält. Eine Position zur Werkstückbearbeitung muss sich in derselben Teilanlage befinden wie das Aktorsignal zur Bearbeitung.

Im Folgenden werden nun die Analysemöglichkeiten hinsichtlich der betrachteten Basis-Messungen diskutiert. Der oben ermittelte Zusammenhang zwischen der Werkstückbearbeitung und der zugehörigen Position wird dabei als gegeben betrachtet.

## 10.2 Zeiten und zeitbezogene Größen

### Gesamtproduktionsdauer, $t_p$ :

Die Gesamtproduktionsdauer wird zur Berechnung von weiteren Qualitätsmerkmalausprägungen benötigt. Da alle Kenngrößen sich nur auf den Zeitraum beziehen können, in dem die zugrundeliegende Ereignissequenz aufgenommen wurde, entspricht die Gesamtproduktionsdauer der Beobachtungsdauer (vgl. auch Formel (7-2)):

$$t_p = K \cdot T_{OPC} \quad (10-1)$$

<sup>47</sup> Eine Ereignissequenz zur MSPN-Generierung der in Abbildung 10-1 gezeigten Werkzeugmaschine ist in Tabelle D-5 im Anhang zu finden

<sup>48</sup> Die zum Beispiel gehörige Sequenz  $E_{Mat}$  ist in Tabelle D-6 im Anhang gezeigt

**Belegungszeit,  $t_r^B$ :**

Die Belegungszeit wird hier durch die Zeit approximiert, in der sich eine technische Ressource nicht im Ruhezustand befindet. Unter der Annahme, dass die Aufnahme von  $E_{obs}$  im Ruhezustand der gesamten Anlage begonnen wurde, ist der Ruhezustand einer Ressource  $R_r$  bestimmt durch den Initialzustand des zugehörigen Maschinenzustandsmodells  $MSPN^r$ . Da jedoch nicht ausgeschlossen werden kann, dass eine Ressource weitere Ruhezustände besitzt außer dem Initialzustand, ist diese Berechnung lediglich als Näherung des Ruhezustandes zu betrachten.

Die Belegungszeit bestimmt sich in dieser Näherung aus der Subtraktion der Dauer die der Initialzustand aktiv war von der Gesamtproduktionsdauer. Ersteres kann berechnet werden aus der Summe aller Produkte der mittleren Aktivierungsdauer und Anzahl an Feuervorgängen, die an denjenigen Transitionen annotiert sind, die Teil des Nachbereiches der Stelle der Initialmarkierung sind:

$$t_r^B = t_p - \sum \mu(J_t) \cdot n(J_t), \forall J_t \in \mathcal{S} \bullet : m_0(\mathcal{S}) = 1 \quad (10-2)$$

Von diesem Wert muss zusätzlich die Zeitdifferenz zwischen dem letzten Ereignis der Ressource und dem letzten Ereignis in  $E_{obs}$  abgezogen werden, wenn angenommen wird, dass die Ressource nach ihrer letzten Tätigkeit und bis zum Ende des Beobachtungszeitraumes sich ebenfalls in Ruhe befunden hat. Ist der Initialzustand nicht Teil eines Zyklus, d.h. das Petri-Netz ist nicht reversibel, vgl. [Abe90], so ist dies ein Indiz dafür, dass ein Hochfahrvorgang der Ressource beobachtet wurde und ein anderer Ruhezustand als der Initialzustand des  $MSPN^r$  existiert. Da zyklisches Verhalten der technischen Ressourcen vorausgesetzt wird, muss der Ruhezustand Teil eines Zyklus von  $MSPN^r$  sein. Wird davon ausgegangen, dass nach einem Hochfahrvorgang zunächst der Ruhezustand erreicht wird, so wird nach dem Zyklus gesucht, der am schnellsten von der Initialmarkierung heraus erreicht werden kann. Zyklen selbst können mittels *T-Invarianten* beschrieben werden [DaAl10]. Ein Vektor  $v$ , der eine T-Invariante beschreibt, kann mithilfe der Inzidenzmatrix mittels Lösung folgender Gleichung bestimmt werden:

$$C \cdot v = 0 \quad (10-3)$$

Somit wird der Ruhezustand der Ressource beschrieben durch die Vorstelle einer Transition, die mindestens einer T-Invariante zugehört und anhand des kürzesten Pfades von der mit der Initialmarkierung besetzten Stelle erreicht werden kann. Die Pfadlänge kann hier anhand der Anzahl an Feuervorgängen, die benötigt werden, um eine Marke von der Initialmarkierung zur besagten Stelle zu transportieren, bestimmt werden. Folgendes Beispiel soll dies verdeutlichen:

Abbildung 10-2 zeigt ein fiktives MSPN (ohne Annotationen), das einen Hochfahrvorgang enthält. Dieser ist gegeben durch die Transition T1. Zyklisches Verhalten des MSPN ist erst erreicht, wenn sich eine Marke nach Feuerung von T1 in P2 befindet.

Dieser Zustand ist dann der Ruhezustand der Ressource. Das Petri-Netz enthält zwei T-Invarianten,  $[0, 1, 1, 1, 0]$  und  $[0, 0, 1, 0, 1]$ . Erstere beschreibt die Feuerfolge  $T2 \rightarrow T3 \rightarrow T4$  und letztere die Feuerfolge  $T3 \rightarrow T5$ . Alle zyklischen Schaltfolgen, die das Netz produzieren kann, entsprechen Linearkombinationen dieser beiden Schaltfolgen. Die Vorstellen aller durch die Invarianten gegebenen Transitionen sind P2, P3 und P4. Aus der Initialmarkierung heraus kann P2 am schnellsten erreicht werden bzw. von P1 ist der kürzeste Pfad zu einer Vorstelle der T-Invarianten über T1 zu P2. Hier muss lediglich eine Transition schalten.

Pfade werden in der Graphentheorie auch als *Weg* bezeichnet [Die00]. Algorithmen zur Bestimmung der Längen von kürzesten Wegen zwischen zwei Punkten eines Graphen beruhen z.B. auf den *Bellmannschen Gleichungen* und finden sich u.a. in [Tur09]. Dieses Vorgehen kann genauso auch für die Belegungszeit der Maschinen,  $t_m^B$ , angewandt werden.

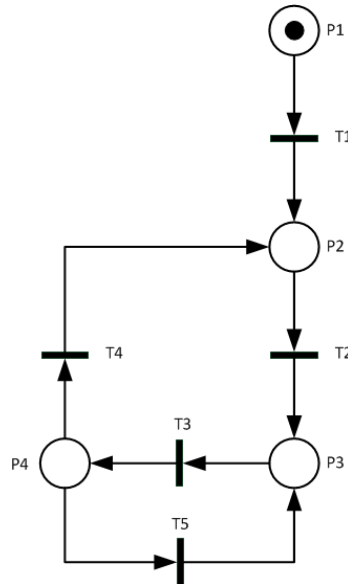


Abbildung 10-2: Beispiel MSPN mit Hochfahrvorgang

#### Effizienz/Dauer der Durchführung von Operationen, $e_{\omega w}^m$ :

Das Verhalten einer Maschine  $M_m$  wird durch ein zugehöriges Maschinenzustandsmodell  $MSPN^m$  beschrieben. Dass es sich bei einem MSPN um eine Maschine handelt, die Werkstücke bearbeitet bzw. Operationen ausführt, ist dadurch zu erkennen, dass der entsprechenden Ressource Signale des Typen *A.WPModify* im Informationsmodell zugewiesen sind. Die Dauer der Durchführung einer Operation entspricht der Differenz zwischen den Zeitstempeln der steigenden Flanke eines solchen Signals und dessen fallender Flanke in der zugehörigen Ereignissequenz  $E^m$ . Der Mittelwert aller so bestimmten Dauern für eine Operation entspricht näherungsweise der gesuchten Effizienz. Es kann allerdings nicht unterschieden werden, ob es sich stets um dieselbe Operation handelt, oder um eine andere. Eine Schätzung kann vorgenommen werden, wenn zwischen zwei durchgeführten Operationen ein Signal vom Typ *A.Toolchange* aufgetreten ist. Da jedoch keine Kenntnis über die Menge an Werkzeugen einer Maschine vorhanden ist, kann keine definitive Aussage darüber getroffen werden, welches Werkzeug derzeit verwendet wird. Somit wird hier jede Werkstück-Bearbeitung durch dieselbe Maschine als dieselbe Operation geschätzt. Eine gängige Art und Weise, Arbeitspläne zu beschreiben, ist die Angabe der geordneten Reihenfolge an durchlaufenen Maschinen, vgl. [KUT93]. Somit ist diese Schätzung durchaus legitimiert.

Alternativ ist es auch möglich die Dauer der Operation als die Zeit zu ermitteln, die sich ein Werkstück in oder an der Maschine befindet. Da die Position der Maschine im MFPN bekannt ist, s.o., entspricht sie der mittleren (evtl. kumulierten, wenn mehr als eine Position in der Maschine angefahren wird) Dauer, die sich ein Werkstück an der Position befindet. So ist es auch möglich, zwischen Werkstücktypen zu unterscheiden, da diese Berechnung für jeden *ident*-Vektor an der Stelle im MFPN vorgenommen werden kann.

### Hauptnutzungszeit der Maschinen, $t_m^{HN}$ :

Die Hauptnutzungszeit einer Maschine  $m$  entspricht der Zeit, in der die Maschine wertschöpfende Aktivitäten durchführt. Eine wertschöpfende Aktivität wird immer dann ausgeführt, wenn ein Signal vom Typen  $A.WPModify$  aktiv ist. Somit entspricht  $t_m^{HN}$  der kumulierten Dauer der 1-Zustände der Signale vom Typen  $A.WPModify$  in der zugehörigen Sequenz  $E^m$ .

### Minimale, mittlere und maximale Produktionsdauer je Werkstücktyp, $t_w^{min}$ , $t_w^\mu$ und $t_w^{max}$

Die Produktionsdauer eines Werkstücktypen kann über die kumulierte Transportdauer vom Anlageneingang (Transition ohne Vorstelle) bis zum Anlagenausgang (Stelle ohne Nachtransition) im MFPN ermittelt werden. Um die Dauer je Werkstücktyp zu ermitteln, wird aus dem MFPN zunächst für jeden **ident**-Vektor ein Teilgraph gebildet. Hierzu werden alle Stellen, Transitionen und Kanten, an denen der entsprechende **ident**-Vektor nicht annotiert ist, entfernt. Zudem werden alle 6-Tupel entfernt, die nicht den entsprechenden **ident**-Vektor enthalten. Abbildung 10-3 zeigt dies an einem Teil des bereits in Abbildung 9-14 gezeigten Ergebnisses der MFPN-Generierung für die Fallstudie. Die so gebildeten Teilgraphen werden im Folgenden als  $MFPN_w$  bezeichnet.

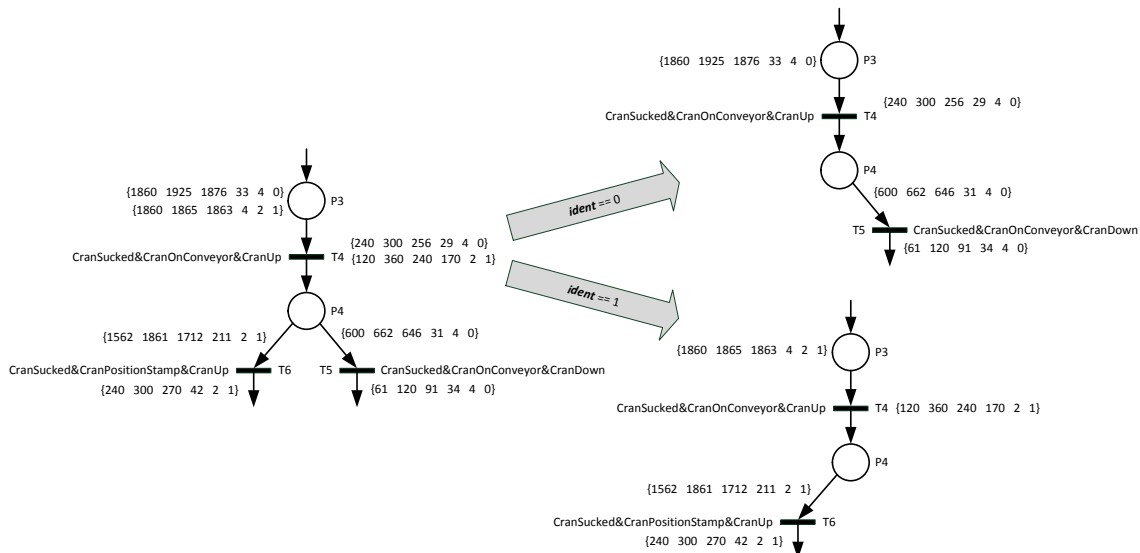


Abbildung 10-3: Beispiel einer Bildung von MFPN-Teilgraphen je **ident**-Vektor

Alle nun folgenden Untersuchungen werden für jeden Werkstücktypen bzw. jeden ermittelten Teilgraphen durchgeführt. Da nicht garantiert werden kann, dass jeder Werkstücktyp nur eine Route durch das System nimmt, müssen die verschiedenen Routen untersucht werden. Für  $t_w^{min}$  und  $t_w^{max}$  werden zunächst die Wege von jedem Anlageneingang zu jedem Anlagenausgang des Teilgraphen ermittelt. Eine Addition aller auf diesen Wegen annotierten Mindesttransportdauern  $d_{min}$  ergibt die kürzeste beobachtete Dauer für die Wege. Die hiervon geringste ermittelte Transportdauer entspricht  $t_w^{min}$ . Graphentheoretisch betrachtet handelt es sich hierbei um die Suche nach dem kürzesten Pfad zwischen den Eingangstransitionen und den Ausgangsstellen unter Berücksichtigung der minimalen Transportdauer als Kantengewicht. Dies kann u.a. mithilfe des *Dijkstra-Algorithmus* ermittelt werden [Tur09]. Für die maximale Produktionsdauer wird das gleiche Verfahren angewandt, allerdings sind die annotierten Maximaltransportdauern zu summieren und das größte Ergebnis zu bestimmen. Die mittlere Produktionsdauer  $t_w^\mu$  wird entsprechend Formel (10-5) bestimmt. Zunächst werden alle annotierten mittleren Transportdauern mit der Anzahl an passierten Werkstücken multipliziert und die Teilergebnisse addiert. Somit

ergibt sich im Dividenden die aggregierte Transportdauer für alle Werkstücke des entsprechenden Typen. Diese wird geteilt durch die die ermittelte produzierte Menge des Werkstücktypen (siehe unten):

$$t_w^\mu = \frac{\sum n(\mathcal{T}_t) \cdot \mu(\mathcal{T}_t) + \sum n(\mathcal{S}_s) \cdot \mu(\mathcal{S}_s) + \sum n(\mathcal{F}_f) \cdot \mu(\mathcal{F}_f)}{PM_w}, \forall \mathcal{T}_t, \mathcal{S}_s, \mathcal{F}_f \in MFPN_w \quad (10-4)$$

### 10.3 Produzierte Menge

Für die produzierte Menge je Werkstücktyp,  $PM_w$ , werden ebenfalls die Teilgraphen  $MFPN_w$  herangezogen. Diese haben eine Menge an Transitionen, die Anlageneingänge für das Werkstück repräsentieren und keine Vorstelle besitzen. Die summierte Menge an annotierten Transportvorgängen an der detektierten Position  $n$  entspricht dabei der Menge an Werkstücken des entsprechenden Typs, die die Anlage betreten haben und somit auch produziert wurden:

$$PM_w = \sum n(\mathcal{T}_t), \forall \mathcal{T}_t \in MFPN_w : \bullet \mathcal{T}_t = \emptyset \quad (10-5)$$

### 10.4 Topologie- und Strukturgrößen

#### Transportverbindungen zwischen Maschinen, $x_{ij}$ :

Der Wert  $x_{ij}$  gibt an, ob zwischen Maschine  $i$  und Maschine  $j$  eine Transportverbindung besteht. Wie in Abschnitt 10.1 bestimmt, ist bekannt, an welchen Transitionen im MFPN sich welche Maschine befindet. Die Existenz einer Transportverbindung zwischen den Maschinen  $i$  und  $j$  kann geprüft werden, indem eine Verbindung in Markenflussrichtung zwischen den Transitionen, die die entsprechenden Positionen repräsentieren, ermittelt wird. Dies kann mithilfe der Tiefensuche (*depth-search-Algorithmus*) auf gerichteten Graphen [Tur09] ermittelt werden. Die Tiefensuche von einem bestimmten Punkt aus spannt einen Baum aller erreichbaren Punkte auf. Wird die Tiefensuche an jener Transition gestartet, die die Position an Maschine  $i$  repräsentiert, kann geprüft werden, ob eine Transition, die die Position an Maschine  $j$  repräsentiert, Teil des Baumes ist, den die Tiefensuche aufspannt.

#### Anzahl Routen je Werkstücktyp $AR_w$

Hier werden wieder die Teilgraphen  $MFPN_w$  des Materialflussmodells je Werkstücktyp betrachtet. Für jede Eingangstransition und jede Ausgangsstelle wird die Anzahl an Pfaden im Graphen bestimmt, z.B. mithilfe der Tiefensuche [Tur09]. Die Summe aller ermittelten Pfade ergibt die Menge an Transportrouten, die der Werkstücktyp nehmen kann.

#### Anzahl Maschinen $|M|$

Die Anzahl Maschinen errechnet sich aus der Anzahl an MSPN, die mindestens ein Ereignis eines Signals vom Typen  $A.WPModify$  enthalten.

## 10.5 Planungs- und Fähigkeitsgrößen

### Prozessfolgen je Werkstücktyp

Prozessfolgen sollen hier genähert als die geordnete Menge durchlaufener Maschinen je Werkstücktyp ermittelt werden. Hierzu wird für jede Route eines Teilgraphen  $MFPN_w$  (s.o.) geprüft, welche Maschinen durchlaufen werden. Jede Maschinenfolge wird als eine Prozessfolge betrachtet. Es sei zu beachten, dass mit dieser Näherung nicht die Gesamtheit aller Operationen bestimmt werden kann, da hier die Annahme zugrunde liegt, dass jede Maschine eine andere Operation durchführt.

### Für Produktfertigung benötigte Operationen je Werkstücktyp $\Omega^w$

Die Menge Operationen, die zur Fertigung eines bestimmten Werkstücktypen  $w$  notwendig ist, kann direkt aus der Menge an Prozessfolgen ermittelt werden. Jede verschiedene Operation, die in mindestens einer oben ermittelten Prozessfolge vorkommt, ist Teil der Menge  $\Omega^w$ .

### Häufigkeit (Gewichtung) der Ausführung von Operationen je Werkstücktyp, $g_\omega$

Wie häufig eine Operation durchgeführt wurde, kann anhand der Anzahl an Feuervorgängen im zugehörigen MSPN bestimmt werden. Hierzu werden die annotierten Anzahlen an Feuervorgängen  $n$  jeder Transition aufsummiert, die eine steigende Flanke eines Signals vom Typen  $A.WPModify$  annotiert haben:

$$g_\omega = \sum n(\mathcal{T}_t), \forall \mathcal{T}_t \in MSPN^m : s_i^\uparrow \subseteq \tilde{e}(\mathcal{T}_t), ST(s_i) \neq auxiliary \quad (10-6)$$

Alternativ kann auch im MFPN ermittelt werden, wie häufig Werkstücke eine Maschine durchlaufen haben.

### Anzahl Produkttypen $|W|$

Die Anzahl an Produkttypen ergibt sich aus der Anzahl unterschiedlicher *ident*-Vektoren eines MFPN. Da nicht allgemein angenommen werden kann, dass jedes Werkstück alle Werkstück identifizierenden Sensoren passiert, kann auch nicht ausgeschlossen werden, dass dieselben Produkttypen unterschiedliche *ident*-Vektoren generieren. Somit kann diese Größe lediglich als Näherung betrachtet werden.

### Häufigkeit der Zuordnung Maschine/Produkttyp $p_{mw}$

Zur Bestimmung der Zuordnungshäufigkeit wird geprüft, wie oft ein Werkstücktyp eine Maschine durchlief. Dies kann anhand der Anzahl an Transportvorgängen  $n$  annotiert an der zur Maschinenposition gehörenden Transition mit entsprechendem *ident*-Vektor ermittelt werden. Diese Anzahl geteilt durch die produzierte Menge des Werkstücktypen ergibt die relative Häufigkeit der jeweiligen Zuordnung.

## 10.6 Zusammenfassung zur Eigenschaftsbestimmung

Obige Analysen erlauben die Bestimmung oder Abschätzung/Näherung einer Vielzahl von Messelementen, die für die Berechnung der Ausprägung von in Abschnitt 3.3. diskutierten Qualitätsmerkmalen herangezogen werden können. Ausgenommen hiervon sind Messelemente, die nicht mithilfe der Ereignissequenzen, des Informationsmodells und der generierten Modelle bestimmt werden können. Tabelle C-1 im Anhang listet alle für die diskutierten Metriken benötigten Messelemente auf. Tabelle 10-1 zeigt auf, welche dieser Messelemente mit dem hier aufgezeigten Ansatz eindeutig bestimmbar

sind (dargestellt mittels „++“), ob sie abgeschätzt oder genähert werden können (+) oder ob sie nicht bestimmbar sind (-).

**Tabelle 10-1: Bestimmbarkeit der in Tabelle C-1 gelisteten Messelemente**

Messelement	Bestimmbar
Belegungszeit, $t_r^B$ :	+
Effizienz/Dauer der Durchführung von Operationen, $e_{\omega w}^m$	+
Hauptnutzungszeit der Maschinen, $t_m^{HN}$	++
Belegungszeit der Maschinen, $t_m^B$	+
Kumulierte Dauer störungsbedingter Unterbrechungen, $t_m^{SU}$	-
Minimale Produktionsdauer, $t_w^{min}$	++
Mittlere Produktionsdauer, $t_w^{\mu}$	++
Maximale Produktionsdauer, $t_w^{max}$	++
Gesamtproduktionsdauer, $t_p$	++
Produzierte Menge, $PM_w$	++
Gutmenge, $GM_w$	-
Anzahl an Gutteilen, $GT_w$	-
Anzahl Prüfteile, $PT_w$	-
Ausschussmenge, $AM_w$	-
Nacharbeitsmenge, $NM_w$	-
Transportverbindungen zwischen Maschinen, $x_{ij}$	++
Anzahl (Transport-)Routen je Werkstück, $AR_w$	++
Anzahl Maschinen, $ M $	++
Anordnung technischer Ressourcen (seriell/parallel)	-
Prozessfolgen (inkl. Anzahl Prozessfolgen $N_w$ )	+
Für Produktfertigung benötigte Operationen $\Omega^v$	+
Durchführbare Operationen (je Maschine und Gesamt), $\Omega^m, \Omega$	-
Häufigkeit (Gewichtung) der Ausführung von Operationen, $g_{\omega}$	+
Anzahl Produkttypen $ W $	+
Häufigkeit der Zuordnung Maschine/Produkttyp $p_{mw}$	++
Mean Time Between Failure, $MTBF_r$	-
Mean Time To Repair, $MTTR_r$	-
Ausfalldichte, $\lambda_r$	-

Insbesondere sind diejenigen Größen nicht bestimmbar, die sich auf Ausfälle, Störungen oder störungsbedingte Unterbrechungen beziehen (z.B. Ausfalldichte), sowie diejenigen Größen, die aus einer Qualitätsprüfung der Produkte resultieren (z.B. Gutmenge).

Welche in Abschnitt 3.3 diskutierten Kennzahlen zur Ermittlung der Qualitätsmerkmalausprägungen mithilfe der durch den Ansatz bestimmbaren Messelemente folglich berechnet werden können, fasst

Tabelle 10-2 zusammen. Auch hier wird zwischen eindeutiger Bestimmbarkeit (alle Messelement der Metrik sind eindeutig bestimmbar, ++), Abschätzung/Näherung (mindestens ein Messelement ist nur durch Abschätzung/Näherung bestimmbar, +) und nicht-Bestimmbarkeit (mind. ein Messelement ist nicht bestimmbar, -) unterschieden.

**Tabelle 10-2: Bestimmbarkeit der Qualitätsmerkmalausprägungen**

Kennzahl (Metrik)	Benötigte Messelemente	Bestimmbar
Beleggrad, $BLG_r$ (3-1)	$t_r^B, t_p$	+
Durchsatz, $DS_w$ (3-2)	$PM_w, t_p$	++
Production rate, $PR_w$ (3-3)	$t_w^\mu$	++
Nutzgrad, $NG_m$ (3-4)	$t_m^{HN}, t_m^B$	+
Effektivität, $Eff$ (3-5)	$PM_w, t_w^\mu, t_m^{HN}$	++
Qualitätsrate, $QR_w$ (3-6)	$GM_w, PM_w$	-
Gesamtanlageneffektivität, $OEE$ (3-7)	$MTBF_r, MTTR_r, \text{Prozessfolgen}, GM_w, PM_w, t_w^\mu, t_m^{HN}$	-
Technischer Nutzgrad, $TN_m$ (3-8)	$t_m^{HN}, t_m^{SU}$	-
Prozessgrad, $PG$ (3-9)	$t_m^{HN}, t_p$	++
First Pass Yield, $FPY_w$ (3-10)	$GT_w, PT_w$	-
Ausschussquote, $AQ_w$ (3-11)	$AM_w, PM_w$	-
Nacharbeitsquote, $NQ_w$ (3-12)	$NM_w, PM_w$	-
Kapazität, $C_w$ (3-13)	$t_w^{min}$	++
Auslastung, $U_w$ (3-14)	$PM_w, t_w^{min}, t_p$	++
Auslastung, $U'_w$ (3-15)	$t_w^{min}, t_w^\mu$	++
Produktivität, $P_m$ (3-16)	$t_m^{HN}, t_p$	++
Verfügbarkeit, $A_r$ (3-17)	$MTBF_r, MTTR_r, \text{Prozessfolgen}$	-
Zuverlässigkeit, $Z_r(t)$ (3-20)	$\lambda_r, \text{Prozessfolgen}$	-
Machine Flexibility, $MF_m$ (3-21)	$\Omega^m, \Omega$	-
Machine Flexibility, $MF'_m$ (3-22)	$g_\omega, e_\omega^m$	+
Machine Flexibility, $MF''_m$ (3-26)	$p_{mw}, \Omega^w, \Omega^m, e_{\omega w}^m$	-
Material Handling Flexibility, $MHF$ (3-27)	$x_{ij},  M $	++
Routing Flexibility, $RF_w$ (3-28)	$AR_w$	++
Operation Flexibility, $OF_w$ (3-29)	$N_w,  \Omega^w $	+
Process flexibility, $PF$ (3-30)	$ M , p_{mw}, \Omega^w$	+

## 11 Fallstudienbasierte Evaluierung des Ansatzes

Dieses Kapitel stellt zwei Fallstudien zur Evaluierung des Ansatzes vor, die im Rahmen vorliegender Arbeit durchgeführt wurden. Ziel der Fallstudien war es, die Anwendbarkeit des vorgestellten Ansatzes aufzuzeigen. Im Fokus standen dabei die Modellgenerierung sowie die Modellanalyse zur Ermittlung der Messelemente. Abschnitt 11.1 beschäftigt sich mit der Durchführung und den Ergebnissen der Anwendung des vorgestellten Ansatzes auf einer Pick and Place Unit (PPU). Diese wurde an der Technischen Universität München speziell für Untersuchungen von evolutionsunterstützenden Ansätzen für industrielle Automatisierung entwickelt. Die zweite Fallstudie (Abschnitt 11.2) wurde am Institut für Automatisierungstechnik der Helmut-Schmidt-Universität in Hamburg durchgeführt. Die verwendete Laboranlage entspricht einer Job-Shop-Fertigung mit mehreren Werkzeugmaschinen und einem Förderband-System. Im Gegensatz zur PPU weist diese Anlage einen komplexeren Materialfluss auf, womit die Fallstudie eine sinnvolle Ergänzung zur Ersteren ist.

Zur Durchführung des Ansatzes wurden die Algorithmen für die Modellgenerierung in der Software *Matlab* (Version R2012a) der Firma *MathWorks* implementiert und ausgeführt. Die Informationsmodelle wurden in tabellarischer Form angelegt<sup>49</sup> und durch die Algorithmen ausgewertet. Zur Aufnahme der Ereignissequenzen wurden die Anlagen mit mehreren Werkstücken gefahren und, parallel dazu, die Daten mittels OPC Data Access (siehe [LIB10]) und einem handelsüblichen Daten-Logger aufgenommen. Die so entstandenen Ereignissequenzen liegen entsprechend in tabellarischer Form gem. Abschnitt 7.2 vor. Die Analysen wurden gem. der in Kapitel 10 dargelegten Algorithmen computergestützt, ebenfalls mit Matlab, durchgeführt.

### 11.1 Anwendung auf eine Pick and Place Unit

Die PPU (siehe Abbildung 11-1) ist eine Benchmarkanlage zur Evaluierung von evolutionsunterstützten Ansätzen. Hierzu wurden vom Lehrstuhl für Automatisierung und Informationssysteme der Technischen Universität München 14 Evolutionsszenarien entwickelt, die Änderungen der Software, Mechanik und/oder Elektrik der Anlage beinhalten. Eine ausführliche Erläuterung der PPU, ihres Verhaltens und aller Evolutionsszenarien ist im zugehörigen technischen Report gegeben [VLF+14]. Faktisch wurden keine Änderungen an der Mechanik oder Elektrik vorgenommen. Stattdessen werden in den Szenarien mechanische und elektrische Komponenten von der Automatisierungssoftware nicht berücksichtigt, sofern sie nicht Teil des Szenarios sind. Es sei erwähnt, dass einige Szenarien keine Änderung im Verhalten der Anlage oder im Signalverhalten zur Folge haben (insbesondere rein elektrische Änderungen). Von den Evolutionsszenarien wurden daher lediglich sechs zur Evaluierung des hier vorgestellten Ansatzes herangezogen. Da die Evolutionsszenarien konsekutiv aufeinander aufbauen, sind dennoch alle Änderungen in der Fallstudie berücksichtigt. Im folgenden Abschnitt werden die Anlage und die für die Fallstudie verwendeten Evolutionsszenarien kurz erläutert. Abschnitt 11.1.2 beschreibt daraufhin die Durchführung der Fallstudie und Abschnitt 11.1.3 diskutiert die Ergebnisse.

---

<sup>49</sup> Im Speziellen wurde das Informationsmodell in einer 2-dimensionalen Zelle hinterlegt. Für jedes Signal der Anlage wurde eine Zeile angelegt. Die erste Spalte enthält den Namen des Signals, die zweite Spalte den Namen der Ressource, dem das Signal zugewiesen ist und die dritte Spalte den Signaltypen.

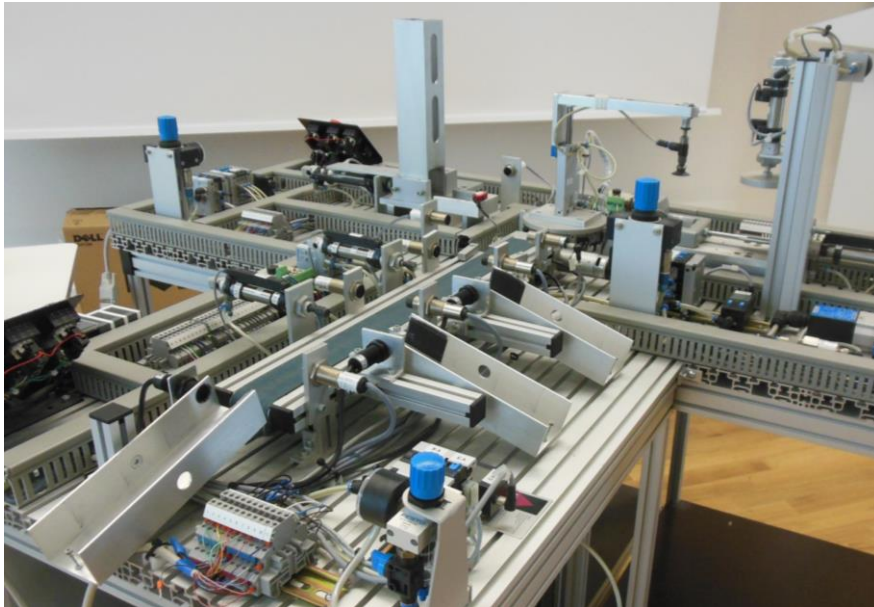


Abbildung 11-1: Pick and Place Unit an der technischen Universität München

### 11.1.1 Aufbau der Anlage und betrachtete Evolutionsszenarien

Die PPU dient der Verarbeitung von zylinderförmigen Werkstücken und besteht insgesamt (je nach Evolutionsszenario) aus vier Anlagenteilen:

1. Einem Werkstückmagazin (*Stack*), das Werkstücke nacheinander in die Anlage bringt (in der Abbildung 11-1 hinten links)
2. Einem Kran (*Crane*), der die Werkstücke aufnehmen und zwischen allen anderen Anlagenteilen transportieren kann (in Abbildung 11-1 hinten mittig)
3. Einem Stempel (*Stamp*), in dem die Bearbeitung der Werkstücke stattfindet (in Abbildung 11-1 hinten rechts)
4. Einem Förderband (*Conveyor*), das die Werkstücke aus der Anlage transportiert (in Abbildung 11-1 vorn).

Alle Anlagenteile sind mit Sensorik und elektrischer sowie pneumatischer Aktorik versehen. Gesteuert wird die gesamte Anlage über eine SPS. Insgesamt enthält die Anlage 48 I/O-Signale<sup>50</sup>. Der Aufbau der Anlage wird im Folgenden konsekutiv anhand der hier betrachteten Evolutionsszenarien erläutert. Die Bezeichnungen der Szenarien sind dabei [VLF+14] entnommen.

#### Szenario Sc1

In diesem Szenario besteht die Anlage aus den beiden Anlagenteilen *Stack* und *Crane* (siehe Abbildung 11-2). Über einen pneumatischen Ausschieber werden die Werkstücke aus dem Magazin geschoben und dem Kran zur Verfügung gestellt. Dieser nimmt die Werkstücke auf und transportiert sie weiter auf eine Ausgangsrampe, die sich an der Position des Förderbandes befindet. Die Ausgangsrampe kann bis zu sechs Werkstücke aufnehmen.

<sup>50</sup> Eine Liste aller in der Anlage vorkommenden I/O-Signale sowie eine Angabe, ab welchem Evolutionsszenario sie verwendet werden, sind in Tabelle E-1 im Anhang zu finden.

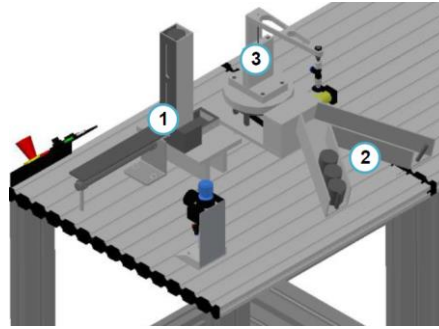


Abbildung 11-2: PPU in Szenario Sc1, bestehend aus dem Eingangsmagazin (1), der Ausgangsrampe (2) und dem Kran (3); Abbildung aus [VLF+14]

### Szenario Sc3

In Sc3 besteht die Anlage zusätzlich zu den Anlagenteilen *Stack* und *Crane* aus dem *Stamp* (Abbildung 11-3). Der Stempel wird dazu genutzt Werkstücke zu stempeln, was einer Bearbeitung der Werkstücke entspricht. Allerdings werden nicht alle Werkstücke bearbeitet. Vielmehr existieren metallische Werkstücke sowie Kunststoff-Werkstücke. Unterschieden werden diese mithilfe eines kapazitiven Sensors, der sich am Werkstückmagazin befindet. Während die Kunststoff-Werkstücke wie zuvor zur Ausgangsrampe transportiert werden, werden die metallischen Werkstücke zum Stempel transportiert, dort bearbeitet und dann zur Ausgangsrampe transportiert. Während des Stempelvorganges wartet der Kran vor dem Stempel, um das Werkstück hinterher abzutransportieren.

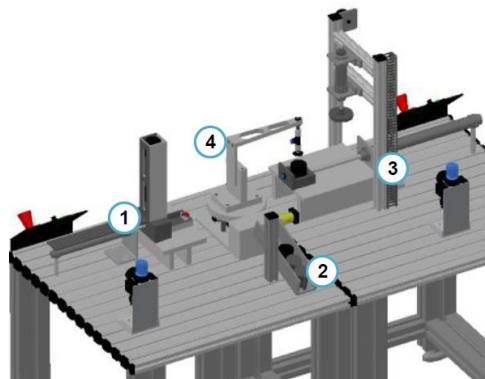


Abbildung 11-3: PPU in Szenario Sc3, bestehend aus dem Eingangsmagazin (1), der Ausgangsrampe (2), dem Stempel (3), und dem Kran (4); Abbildung aus [VLF+14]

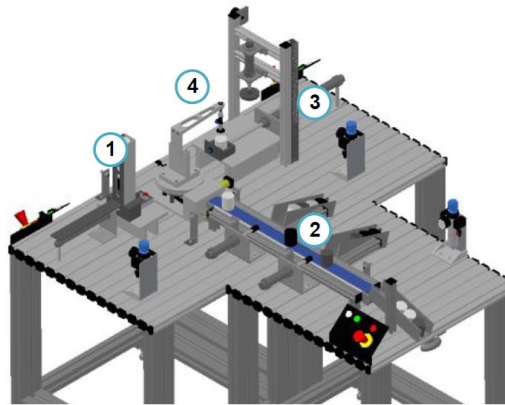
### Szenario Sc5

Dieses Szenario gleicht im wesentlichen Sc3. Allerdings wird, um den Durchsatz zu erhöhen, während der Stempelung eines metallischen Werkstückes geprüft, ob ein Kunststoff-Werkstück am Magazin zur Verfügung steht. Ist dies der Fall, so wird das Kunststoff-Werkstück vom Magazin zur Ausgangsrampe transportiert, während das metallische Werkstück gestempelt wird. Danach wird das fertig gestempelte metallische Werkstück vom Stempel zur Ausgangsrampe transportiert.

### Szenario Sc10

In Sc10 kommt zu den vorigen Anlagenteilen, anstelle der Ausgangsrampe, das Förderband (*Conveyor*) zum Einsatz. Entlang des Förderbandes befinden sich zwei Ausgangsrampen sowie eine weitere am Ende des Förderbandes (siehe auch Abbildung 11-4). Jede der Ausgangsrampen kann exakt vier Werkstücke aufnehmen. Über pneumatische Ausschieber können Werkstücke in die seitlichen Ausgangsrampen vom

Förderband gestoßen werden. Zunächst wird die Rampe am Ende des Förderbandes gefüllt, danach die hintere Ausgangsrampe senkrecht zum Förderband und zuletzt die vordere Ausgangsrampe senkrecht zum Förderband. Zudem wird nun zwischen weißen und schwarzen Kunststoff-Werkstücken unterschieden. Die schwarzen Kunststoff-Werkstücke werden wie in den vorangegangenen Szenarien behandelt. Die weißen Kunststoff-Werkstücke werden, wie die metallischen Werkstücke, vom Stempel bearbeitet. Um zwischen den schwarzen und den weißen Werkstücken unterscheiden zu können, befindet sich nun am Magazin zusätzlich ein optischer Sensor, der die Helligkeit der Werkstücke misst und ein entsprechendes binäres Signal (hell/dunkel) generiert.



**Abbildung 11-4: PPU in Szenario Sc10, bestehend aus dem Eingangsmagazin (1), dem Förderband (2), dem Stempel (3), und dem Kran (4); Abbildung aus [VLF+14]**

### Szenario Sc11

Vom Aufbau her entspricht dieses Szenario dem Vorigen. Allerdings werden die Werkstücke nun nicht mehr der Reihenfolge nach in die Ausgangsrampen befördert, sondern entsprechend ihres Typen. Weiße Kunststoff-Werkstücke werden in die Ausgangsrampe am Ende des Förderbandes sortiert, schwarze Kunststoff-Werkstücke in die vordere Rampe senkrecht zum Förderband und metallische Werkstücke in die hintere Rampe senkrecht zum Förderband. Um die Werkstücke am Förderband unterscheiden zu können, werden weitere optische (Unterscheidung hell/dunkel) und kapazitive (Unterscheidung Kunststoff/Metall) Sensoren genutzt. Je einer befindet sich vor jeder Rampe senkrecht zum Förderband sowie aus Redundanzgründen ein zusätzlicher Helligkeitssensor vorn am Förderband. Es werden somit, verglichen mit Szenario Sc10, fünf weitere Sensoren verwendet.

### Szenario Sc12

Dieses letzte Szenario unterscheidet sich vom Vorigen ausschließlich in der Art und Weise, wie die Werkstücke in den Ausgangsrampen sortiert werden. In Sc12 wird das Ziel verfolgt, von jedem Werkstücktypen ein Werkstück in jede Ausgangsrampe zu transportieren. Ist ein Werkstück von einem bestimmten Typen in keiner Rampe vorhanden, so wird es zunächst in die Rampe am Ende des Förderbandes sortiert. Befindet sich dort schon ein Werkstück des aktuellen Typen, so wird es in die hintere Rampe senkrecht zum Förderband sortiert. Befindet sich dort ebenfalls schon ein Werkstück des Typen, wird das aktuelle Werkstück in die vordere Rampe senkrecht zum Förderband sortiert. Zusätzlich kommt ein weiterer Helligkeitssensor vor der Rampe am Ende des Förderbandes zum Einsatz.

#### 11.1.2 Durchführung der Fallstudie

Zur Durchführung der Fallstudie wurde zunächst ein Informationsmodell für die PPU angelegt. Hierbei wurden alle I/O-Signale berücksichtigt, so dass ein Informationsmodell über alle Szenarien genutzt wer-

den konnte. Die Equipmenthierarchie teilt die PPU in die vier technischen Ressourcen *Stack*, *Crane*, *Stamp* und *Conveyor*. Entsprechend ihrer Zugehörigkeit zu diesen technischen Ressourcen wurden die Signale dem Equipment im Informationsmodell zugewiesen. Zudem wurde jedem Signal entsprechend seiner Bedeutung ein Signaltyp gem. Abschnitt 7.3 zugewiesen<sup>51</sup>. Die Erstellung des Informationsmodells war aufwandsarm (weniger als eine Stunde) und konnte problemlos mithilfe der Kenntnis über die Bedeutung der verschiedenen Signale erstellt werden. Die Bedeutung der Signale konnte der vorhandenen Signalliste (vgl. Tabelle E-1 im Anhang) entnommen werden.

Die PPU wurde in jedem oben beschriebenen Szenario betrieben und die Daten aufgenommen. Der OPC-Server-Zyklus bei der Aufnahme der Daten betrug 60ms. Tabelle 11-1 fasst die Rahmenbedingungen der Durchführung der Fallstudie zusammen.

**Tabelle 11-1: Rahmenbedingungen bei der Durchführung der Fallstudie für die Evolutionsszenarien**

Evolutionsszenario		Sc1	Sc3	Sc5	Sc10	Sc11	Sc12
<b>Rahmenbedingung</b>							
<b>#Werkstücke</b>	<b>Kunststoff-weiß</b>	2	2	2	2	2	2
	<b>Kunststoff-schwarz</b>	2	2	2	3	3	3
	<b>Metall</b>	2	2	2	3	3	3
<b># Signale</b>		18	32	32	42	48	48
<b># Ereignisse</b>		215	310	324	601	609	625
<b>Parameter</b>	<b><math>T_{thresh}</math></b>	60	60	60	60	60	60
	<b>cost</b>	1500	300	350	300	300	1000
	<b>maxDistance</b>	3000	3000	3000	2300	3000	3000
	<b>maxOverlap</b>	300	300	300	700	650	750

In der Ausführung der Szenarien Sc1 bis Sc5 wurden jeweils sechs Werkstücke durch die Anlage verarbeitet und in den übrigen Szenarien jeweils acht Werkstücke (siehe Tabelle 11-1). Grundsätzlich wurde angestrebt, so viele Werkstücke wie möglich durch die Anlage bearbeiten zu lassen, um das Verhalten möglichst vollständig beobachten zu können. In den ersten Szenarien ist die Anzahl der Werkstücke jedoch durch die Kapazität der Ausgangsrutsche begrenzt und in den späteren Szenarien durch die Menge der vorhandenen Werkstücke. Zeile vier der Tabelle zeigt die Anzahl an I/O-Signalen auf, die für das entsprechende Szenario relevant sind. Zeile fünf benennt die Anzahl der Ereignisse, die bei der Durchführung des Szenarios aufgenommen wurden.

Die letzten vier Zeilen in Tabelle 11-1 geben an, wie die Lernalgorithmen parametrisiert wurden. Der erste Parameter,  $T_{thresh}$ , gibt an, innerhalb welcher Zeitspanne Ereignisse zum Lernen von MSPN-Modellen zusammengefasst werden (vgl. Abschnitt 8.2). Dieser Parameter wurde stets der eingestellten OPC-Server-Zyklusdauer von 60 ms gleichgesetzt. Der Parameter *cost* entspricht der maximalen Standardabweichung, die ein gefundenes Cluster bei der Zuweisung von Ereignissen zu Werkstückinstanzen beim Lernen von MFPN-Modellen aufweisen darf (vgl. Abschnitt 9.2). Die letzten beiden Parameter wurden eingeführt, um den MFPN-Lernalgorithmus für die praktische Durchführung robuster zu gestalten. Die Notwendigkeit hierfür ergab sich zum einen daraus, dass insbesondere aufgrund der geringen Anzahl an Werkstücken in einigen Fällen zufällige zeitstabile Zusammenhänge zwischen Ereignissen erkannt wurden. Dies führte dazu, dass falsche Zuordnungen von Ereignissen zu Werkstückinstanzen vorgenommen wurden. Um die Wahrscheinlichkeit einer solchen Zuordnung zu verringern, wurde der Parameter

<sup>51</sup> Das vollständige Informationsmodell in tabellarischer Darstellung ist in Tabelle E-2 im Anhang aufgezeigt.

*maxDistance* eingeführt, der angibt, welcher zeitliche Abstand maximal zwischen zwei Ereignissen vorherrschen darf, damit diese auf einen zeitstabilen Zusammenhang überprüft werden. So wird vermieden, dass ein zufällig zeitstabiler Zusammenhang weit entfernter Signale ermittelt wird. Außerdem konnte somit die notwendige Rechendauer reduziert werden. Zum anderen wurde der Parameter *maxOverlap* eingeführt. Dieser gibt den maximalen Zeitraum an, den sich Signale überlappen dürfen, um noch von derselben Werkstückinstanz ausgelöst worden sein zu können. Überlappen bedeutet hier, dass die Signale sich zeitgleich im 1-Zustand befinden. Eine Überlappung kann dort stattfinden, wo eine Werkstückübergabe stattfindet. Am Beispiel der PPU kommt sie u.a. vor, wenn der Kran ein Werkstück vom Magazin aufnimmt. Für einen gewissen Zeitraum (ca. 100-200ms) wird das Werkstück noch am Magazin detektiert, jedoch auch schon als vom Kran aufgenommen erkannt. Da die Anlage stochastischen Störeinflüssen, wie bspw. Schlupferscheinungen, unterliegt und die Zeitstempelung auch teilweise aufgrund von Rechenzeiten und Buslasten stochastisch schwankt, mussten die genannten Parameter für die Ereignissequenzen, die sich aus der Durchführung der Szenarien ergaben, stets entsprechend eingestellt werden, um korrekte Ergebnisse bei der Modellgenerierung zu erhalten.

### 11.1.3 Diskussion der Ergebnisse

Für alle dargestellten Szenarien konnten MSPNs für alle relevanten Anlagenteile und je ein MFPN generiert werden. Eine Sichtprüfung der Modelle ergab, dass sie das Verhalten der Anlage im jeweiligen Szenario, abzüglich einer Toleranz in den Zeitwerten, die sich aus ungenauer Zeitstempelung ergab, korrekt wiedergeben. Zur Abschätzung der Komplexität der generierten Petri Netze gibt Tabelle 11-2 Eckdaten der MSPNs und des MFPN je Szenario an.

Tabelle 11-2: Eckdaten der generierten Petri-Netze

		Sc1	Sc3	Sc5	Sc10	Sc11	Sc12
<b>MSPN<sup>Crane</sup></b>	$ \mathcal{T} $	34	30	37	45	41	54
	$ \mathcal{S}_{mach} $	21	18	21	25	23	29
	$ \mathcal{S}_{sig} $	20	22	22	22	22	22
<b>MSPN<sup>Stack</sup></b>	$ \mathcal{T} $	11	13	19	28	22	22
	$ \mathcal{S}_{mach} $	8	9	11	15	13	12
	$ \mathcal{S}_{sig} $	8	10	10	12	12	12
<b>MSPN<sup>Stamp</sup></b>	$ \mathcal{T} $	-	13	16	18	18	16
	$ \mathcal{S}_{mach} $	-	12	13	13	14	13
	$ \mathcal{S}_{sig} $	-	16	16	16	16	16
<b>MSPN<sup>Conveyor</sup></b>	$ \mathcal{T} $	-	-	-	15	24	32
	$ \mathcal{S}_{mach} $	-	-	-	11	19	23
	$ \mathcal{S}_{sig} $	-	-	-	16	26	28
<b>MFPN</b>	$ \mathcal{T} $	5	12	12	18	17	15
	$ \mathcal{S} $	5	11	11	16	16	14

Für jedes MSPN ist die Anzahl an Transitionen  $|\mathcal{T}|$ , die Anzahl an Stellen des auf Maschinenverhalten bezogenen Teils  $|\mathcal{S}_{mach}|$  sowie die Anzahl an Stellen des auf Signalverhalten bezogenen Teils  $|\mathcal{S}_{sig}|$  gegeben. Es sei anzumerken, dass  $|\mathcal{S}_{sig}|$  stets der zweifachen Anzahl an beobachteten Signalen entspricht. Dies hat den Hintergrund, dass der Algorithmus exakt zwei Stellen für jedes Signal generiert, eine für dessen 1-Zustand und eine für dessen 0-Zustand. Auffällig bei Betrachtung der Eckdaten ist, dass die Anzahl an Stellen und Transitionen sich auch ändert, wenn sich das Verhalten der zugehörigen Resource nicht geändert hat (erkennbar z.B. an der nicht konstanten Anzahl für den Stempel). Dies lässt

sich dadurch erklären, dass die Kombination von Ereignissen innerhalb des Schwellwertes  $T_{thresh}$  in den unterschiedlichen Szenarien nicht stets die gleichen Ereignisse zusammenführte. Der Grund hierfür ist, dass die Ungenauigkeiten der Zeitstempel in den unterschiedlichen Szenarien und auch in den unterschiedlichen Durchläufen schwankten. Es sei jedoch erwähnt, dass das Verhalten dennoch stets korrekt modelliert wird, sich allerdings unterschiedliche kombinierte Ereignisse  $\tilde{e}_j^r(k)$  ergaben.

Die Modelle konnten mittels der in Kapitel 10 vorgestellten Algorithmen analysiert werden<sup>52</sup>. Im Folgenden werden einige auserwählte Ergebnisse und, anhand derer, Eigenschaften der Methode und der implementierten Algorithmen diskutiert.

Der Stempel wurde in allen Szenarien, in denen er zum Einsatz kam, als Werkstück bearbeitende Maschine identifiziert, da er ein Signal (*Actuator\_StampPusher*) vom Typen *A.WPModify* beinhaltet. Das Signal, das die Position im Stempel angibt, konnte anhand der Zugehörigkeit zum Stempel im Informationsmodell ermittelt werden. Die Dauer der Operationen wurde auf zwei Arten bestimmt (vgl. Abschnitt 10.1). Zum einen anhand der mittleren Dauer, die sich das Signal *Actuator\_StampPusher* im 1-Zustand befand (ermittelt aus der zugehörigen Ereignissequenz). Zum anderen anhand der Dauer, in der sich ein Werkstück im Stempel befand (ermittelt aus dem MFPN). Diese Werte liegen jedoch um den Faktor 10 bis 20 auseinander. Dies liegt daran, dass das Werkstück im Stempel zunächst mithilfe des Schlittens zum Stempel herangeführt werden muss. Erst dann findet die tatsächliche Stempelung statt. Da das Werkstück nach der Stempelung teilweise auf den Kran warten muss, da er in der Zwischenzeit ein anderes Werkstück transportiert (ab Sc5, siehe oben), ist die Dauer, die sich ein Werkstück im Mittel im Stempel befindet, auch davon abhängig, in welcher Reihenfolge die Werkstücke in die Anlage kommen. Der Wert ist demnach stets zu hoch. Vorteilhaft an der Bestimmung mittels MFPN ist jedoch, dass die Dauer nach Werkstücktypen unterschieden werden kann, da diese Information nur im MFPN enthalten ist. Als Schlussfolgerung kann ausgesagt werden, dass die Analyseergebnisse, die der Ansatz liefert, durch anlagenkundige Personen kritisch hinterfragt werden sollten, um solche Ungenauigkeiten in den Ergebnissen zu erkennen und in der Bewertung zu berücksichtigen.

Es konnte ermittelt werden, dass der Kran in allen Szenarien eine sehr hohe Belegungszeit aufweist, die nahe an die gesamte Produktionsdauer reicht. Dies resultiert aus der Tatsache, dass der Kran fast die gesamte Zeit mit dem Transportieren von Werkstücken beschäftigt ist. Da die Analyse hinsichtlich der Belegungszeit lediglich eine Ruhelage einer Ressource berücksichtigt, nämlich die Stelle eines MSPN, die den kürzesten Weg zur initial markierten Stelle aufweist und Teil eines Zyklus ist (vgl. Abschnitt 10.1), werden hier keine Wartezeiten berücksichtigt. In Sc3 wartet der Kran bspw. auf die Fertigstellung einer Stempelung, bevor er ein Werkstück vom Stempel zur Ausgangsrampe transportiert. Diese Zeit fließt mit in die Belegungszeit ein. Wie in Sc5 eingeführt, könnte diese Zeit jedoch auch anderweitig genutzt werden, womit sie nicht Teil der Belegungszeit sein muss. Auch hier zeigt sich, dass mit Ungenauigkeiten in den Analyseergebnissen zu rechnen ist. Mit dieser Kenntnis eignen sie sich dennoch zur Bewertung sich ändernden Verhaltens.

Der praktische Nutzen des Ansatzes lässt sich anhand der Analyseergebnisse von Sc3 und Sc5 verdeutlichen. In Sc5 soll der Durchsatz erhöht werden, indem der Kran ein Kunststoff-Werkstück transportiert, während ein metallisches gestempelt wird. Es zeigt sich jedoch, dass die maximale Produktionsdauer und auch die mittlere Produktionsdauer von metallischen Werkstücken tatsächlich ansteigen. Dies liegt daran, dass metallische Werkstücke nun im Stempel warten müssen, bis der Transport eines Kunststoff-

---

<sup>52</sup>Die genauen Werte, die sich aus den Analysen ergaben, sind im Anhang in den Tabellen E-3 – E-8 zu finden.

Werkstückes abgeschlossen ist. Dies lässt sich auch daran erkennen, dass die mittlere Dauer, die sich ein Werkstück im Stempel befindet, von 6 Sekunden auf ca. 12 Sekunden ansteigt. Auch die Gesamtproduktionsdauer ist bei gleichbleibendem Produktmix um knapp 10% angestiegen. Ursprünglich war angenommen worden, dass eine parallele Bearbeitung den Durchsatz erhöhen würde. Die Analyse konnte jedoch einen Seiteneffekt aufdecken, der dem entgegenwirkt.

In den Szenarien Sc3 und Sc5 konnten zwei Werkstücktypen anhand des einen Signals vom Typ *S.WPIIdentify* unterschieden werden. Dieses ist das Signal *Sensor\_MagazinCapacitiveSensor*, das vom kapazitiven Sensor am Magazin stammt. Somit können im MFPN metallische Werkstücke (*ident*-Vektor [1]) und Kunststoff-Werkstücke (*ident*-Vektor [0]) unterschieden werden. Sc10 führt ein weiteres Identifikationssignal ein, *Sensor\_MagazinOpticalSensor*. Im MFPN finden sich entsprechend Werkstücktypen mit *ident*-Vektoren [00] (weiß, Kunststoff), [01] (schwarz, Kunststoff) und [11] (metallisch) wieder. In Sc11 werden weitere Identifikationssensoren am Förderband verwendet. Diese sind je ein kapazitiver und ein Helligkeitssensor an den Ausschibern des Förderbandes sowie ein Helligkeitssensor vorne am Förderband. Da Werkstücke des gleichen Typen in Sc11 stets die gleichen Sensoren passieren, weil sie durchweg in die gleiche Ausgangsrutsche transportiert werden, ergeben sich wieder exakt drei *ident*-Vektoren, jedoch mit sieben Signalzuständen. Im folgenden Szenario Sc12 wird nicht nur ein weiterer Identifikationssensor eingeführt (Helligkeitssensor am Ende des Förderbandes), sondern jeder Werkstücktyp wird auch in jede Ausgangsrutsche befördert. Da einige Werkstücke nun einige Sensoren nicht auslösen, weil sie den entsprechenden Sensor einfach nicht passieren, ergeben sich hier sechs verschiedene *ident*-Vektoren, obwohl nur drei Werkstücktypen existieren. Entsprechend müssen, für eine korrekte Interpretation der Analyseergebnisse, die Ergebnisse für unterschiedliche *ident*-Vektoren, die jedoch den gleichen Werkstücktypen beschreiben, manuell aggregiert werden.

Es hat sich gezeigt, dass der beschriebene Ansatz erfolgreich auf die Fallstudie angewandt werden konnte. Mit geringem Aufwand konnten die Merkmalsausprägungen ermittelt werden. Mithilfe dieser ist es möglich, die vorgenommenen Evolutionsschritte zu bewerten. Allerdings hat sich auch gezeigt, dass eine Interpretation der Analyseergebnisse durch anlagenkundiges Personal notwendig ist. Dies hat sich insbesondere bei den Ergebnissen zu Sc12 gezeigt, in denen verschiedene *ident*-Vektoren für dieselben Werkstücktypen ermittelt wurden.

## 11.2 Anwendung auf eine Job-Shop Fertigung

In diesem Abschnitt wird nun die zweite Fallstudie zur Evaluierung des Ansatzes vorgestellt. Bei der verwendeten Anlage handelt es sich um eine Modellanlage, die eine Job-Shop-Fertigung darstellt (Abbildung 11-5). Diese befindet sich am Institut für Automatisierungstechnik der Helmut-Schmidt-Universität (HSU) in Hamburg.



Abbildung 11-5: Modellanlage des Instituts für Automatisierungstechnik an der HSU in Hamburg

Zur Anwendung des Ansatzes wurden für die Anlage drei verschiedene Evolutionsszenarien betrachtet. Diese unterscheiden sich in erster Linie im Routing der Werkstücke sowie in den Prozessfolgen, mit denen die Werkstücke bearbeitet werden. Da solche Änderungen an der PPU nicht betrachtet werden konnten, eignet sich diese Fallstudie zur Ergänzung der bereits diskutierten Evaluierung an der PPU. Die Gliederung dieses Abschnitts entspricht dabei der des vorangegangenen Abschnittes.

### 11.2.1 Aufbau der Anlage und betrachtete Evolutionsszenarien

Die Anlage besteht aus drei Werkzeugmaschinen sowie einem Transportsystem, das Werkstücke mittels Förderbändern und Drehtischen zwischen den Maschinen transportiert (Abbildung 11-6). Sie dient der Bearbeitung unterschiedlicher Werkstücke. Insgesamt existieren fünf unterschiedliche Werkstücktypen, die sich in ihrer Farbe unterscheiden: rot, blau, grün, gelb und holzfarben (natur). Die Anlage ist unterteilt in fünf Teilanlagen, die im Folgenden mit *Slave 1* - *Slave 5* bezeichnet werden. Jede Teilanlage wird von einer SPS gesteuert. Eine weitere SPS dient als Kopfsteuerung, die dafür zuständig ist, Daten zwischen den Teil-Steuerungen auszutauschen. Insgesamt besitzt die Anlage 146 I/O-Signale, von denen letztlich 125 relevant für die Fallstudie sind<sup>53</sup>. Die nicht relevanten Signale dienen der Ansteuerung von Signalleuchten oder werden in den betrachteten Szenarien nicht verwendet. Im Folgenden werden die einzelnen Anlagenteile und darauf aufbauend die drei Evolutionsszenarien, die zur Evaluierung des Ansatzes herangezogen wurden, kurz erläutert.

Slave 1 besteht aus dem Anlageneingang in Form eines Registerlager-Bandes, in dem die Werkstücke in die Anlage eingebracht werden, sowie drei Förderbändern. Hier werden auch die Werkstücktypen ermittelt. Hierzu befinden sich am Registerlager-Band zwei Reed-Kontakte, je einer auf der linken Seite und einer auf der rechten Seite des Förderbandes. An den Werkstücken sind, je nach Typ, Nägel auf der Unterseite angebracht, die Signale an den Reed-Kontakten auslösen. Die Anzahl der Nägel dient der Unterscheidung der Werkstücktypen. Tabelle 11-3 listet auf, wie viele Nägel zur Identifizierung der Werkstücktypen angebracht sind.

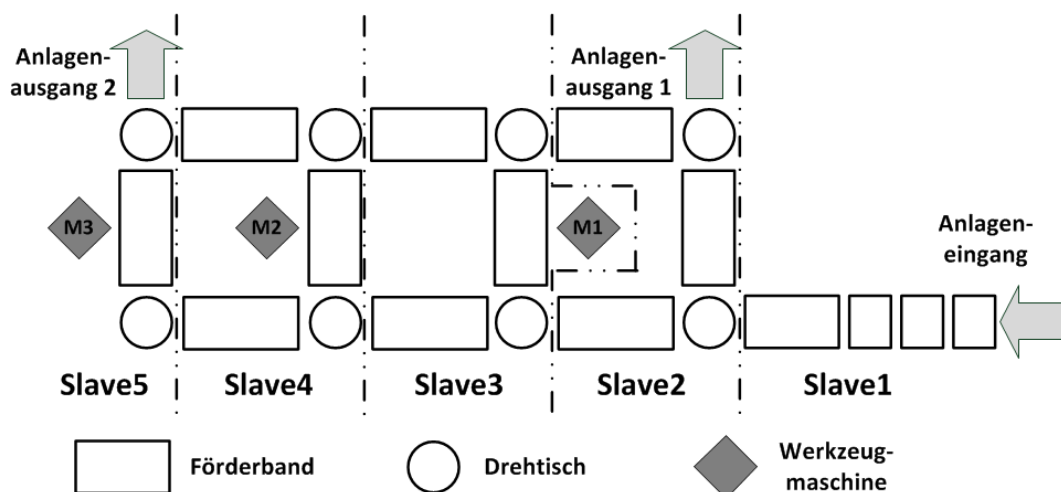


Abbildung 11-6: Übersicht über die Fallstudienanlage

<sup>53</sup> Eine Liste aller I/O-Signale inkl. einer Beschreibung dieser ist im Anhang in Tabelle F-1 zu finden. Zudem zeigen die Prinzipskizzen in den Abbildungen F1-F5 des Anhangs den genauen Aufbau der Anlage sowie die Einordnung der I/O-Signale in die Anlage.

Tabelle 11-3: Anzahl angebrachter Nägel zur Identifizierung der Werkstücktypen

Anzahl Nägel	
Natur	0
Gelb	1
Rot	2
Grün	3
Blau	4

In Slave 2 befinden sich zwei Drehtische sowie drei Förderbänder. Von einem der Drehtische (oben in Abbildung 11-6) können Werkstücke zudem von einem Kran, der hier nicht weiter als Teil der Anlage betrachtet wird, abgeholt und somit aus der Anlage entfernt werden. Ein weiterer Anlagenausgang befindet sich am oberen Drehtisch in Slave 5. Die Werkzeugmaschinen befinden sich in den Teilanlagen Slave 3 bis Slave 5. An ihnen wird die Bearbeitung der Werkstücke vorgenommen.

Die drei betrachteten Evolutionsszenarien werden im Folgenden vorgestellt:

### Szenario Sc1

Im Ausgangsszenario Sc1 werden die Werkstücke gem. dem in Abbildung 11-7 gezeigten Routing durch die Anlage transportiert. Naturfarbene Werkstücke werden nicht bearbeitet, sondern am Anlagenausgang 1 wieder aussortiert. Alle anderen Werkstücktypen werden nacheinander an den Maschinen M1, M2 und dann M3 bearbeitet.

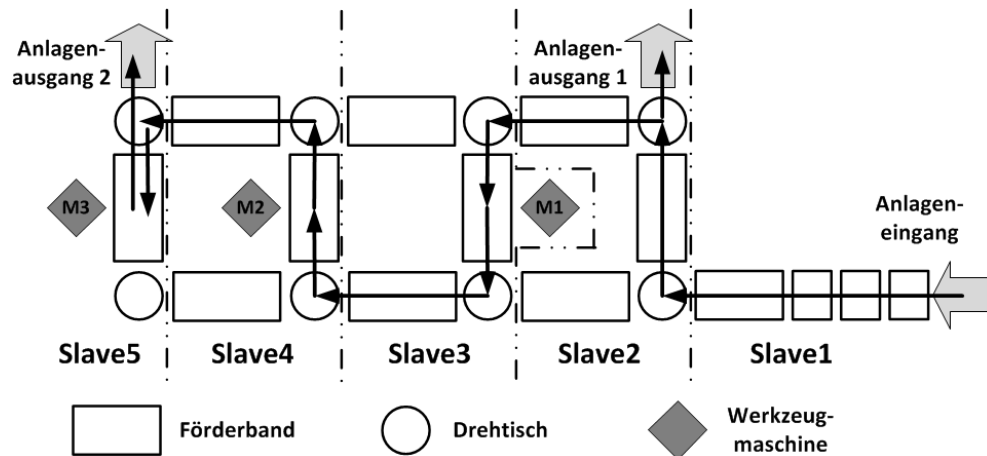


Abbildung 11-7: Werkstück-Routing in Sc1

Nach der letzten Bearbeitung an M3 werden die Werkstücke dann mittels Anlagenausgang 2 aus der Anlage transportiert. Die Anlagensteuerung ist so programmiert, dass sich in jeder Teilanlage lediglich ein Werkstück zurzeit aufhalten kann. Soll ein Werkstück aus einer Teilanlage in eine nachfolgende Teilanlage transportiert werden, in der sich bereits ein Werkstück befindet, so wird das Werkstück solange in der Teilanlage gehalten, bis die nachfolgende Teilanlage wieder frei wird.

### Szenarien Sc2 und Sc3

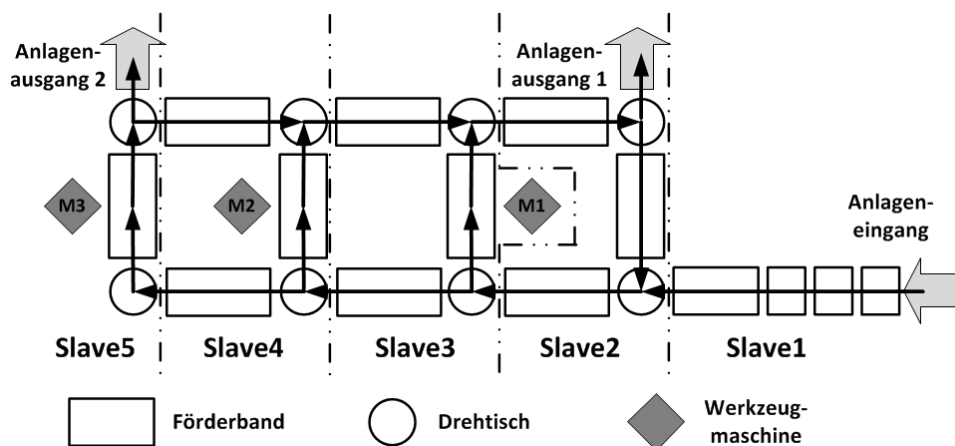
In den auf Sc1 aufbauenden Szenarien Sc2 und Sc3 wurde in erster Linie die Bearbeitungsfolge einzelner Werkstücktypen angepasst (siehe Tabelle 11-4 für eine zusammenfassende Darstellung). Um eine mög-

lichst frei wählbare Bearbeitungsreihenfolge zu ermöglichen, musste hierzu zunächst das Routing der Werkstücke angepasst werden (siehe Abbildung 11-8).

**Tabelle 11-4: Zusammenfassende Darstellung der Bearbeitungsreihenfolge in den Szenarien**

Szenario	Werkstücktyp	Bearbeitung 1	Bearbeitung 2	Bearbeitung 3
Sc1	rot	M1	M2	M3
	blau	M1	M2	M3
	grün	M1	M2	M3
	gelb	M1	M2	M3
	natur	-	-	-
Sc2	rot	M1	M2	M3
	blau	M1	M2	M3
	grün	M1	M2	M3
	gelb	M1	M2	M3
	natur	M3	-	-
Sc3	rot	M1	M2	M3
	blau	M1	M2	M3
	grün	M1	M2	M3
	gelb	M3	M2	M1
	natur	-	-	-

Im angepassten Routing werden die Werkstücke auf den unteren Förderbändern und Drehtischen stets von rechts nach links transportiert und auf den oberen Förderbändern und Drehtischen von links nach rechts. An den unteren Drehtischen in Slave 3, Slave 4 und Slave 5 wird geprüft, ob die nächste Bearbeitung durch die in der entsprechenden Teilanlage enthaltene Maschine stattfinden soll. Ist dies der Fall, wird das Werkstück nach oben zur Maschine befördert. An den oberen Drehtischen in Slave 5 und Slave 2 wird geprüft, ob die Bearbeitung des Werkstückes beendet ist und das Werkstück aus der Anlage heraus transportiert werden kann. Somit werden die Werkstücke stets im Uhrzeigersinn durch die Anlage transportiert. Da dies zu erheblich erhöhten Transportwegen führen kann, wurde zusätzlich der Durchsatz je Teilanlage erhöht. Dazu wurde jede Teilanlage dazu befähigt, mehr als ein Werkstück aufzunehmen. Es gilt hier nur noch, dass jedes Förderband und jeder Drehtisch nur ein Werkstück zurzeit aufnehmen kann.



**Abbildung 11-8: Werkstück-Routing in Sc2 und Sc3**

Sc2 und Sc3 unterschieden sich darin, welche Bearbeitungsreihenfolgen im Vergleich zu Sc1 geändert wurden. Zudem wurden die Bearbeitungsdauern der Maschinen M2 und M3 verringert. In Sc2 werden

nun auch naturfarbene Werkstücke bearbeitet. Dies geschieht durch Maschine M3 in Slave 5. In Sc3 werden naturfarbene Werkstücke weiterhin nicht bearbeitet. Jedoch ändert sich hier die Bearbeitungsreihenfolge von gelben Werkstücken. Diese werden nun in fester Reihenfolge durch M3, dann M2 und dann M1 bearbeitet. Die Reihenfolge ist also genau umgekehrt zur vorigen in Sc1.

### 11.2.2 Durchführung der Fallstudie

Wie für die PPU wurde das Informationsmodell für die Job-Shop-Anlage tabellarisch angelegt. Auch hier war der Aufwand moderat. Das Informationsmodell konnte in weniger als drei Stunden erstellt werden. Die Equipmenthierarchie teilt die Anlage zunächst in die fünf Teilanlagen und dann in die einzelnen technischen Ressourcen. Eine technische Ressource entspricht dabei stets einem Förderband, einem Drehtisch oder einer Werkzeugmaschine. Somit ergaben sich 28 technische Ressourcen. Signale, die keiner dieser Ressourcen zugewiesen werden konnten, z.B. zur Ansteuerung einer Signalleuchte in Slave 1, wurden dem übergeordneten Equipment (Slave 1 – Slave 5) zugewiesen. Allerdings sind diese auch nicht weiter betrachtet worden, da sie irrelevant für den Fertigungsprozess sind<sup>54</sup>.

Da alle für den Ansatz relevanten Daten in der Kopfsteuerung auflaufen, wurden sie von dort aufgenommen. Auch hier wurde ein OPC-Server-Zyklus von 60 ms gewählt, der jedoch aufgrund der Vielzahl an aufkommenden Daten meist nicht eingehalten werden konnte. Somit schwankte der Zyklus zwischen 60 ms und 100 ms. Da an dieser Anlage sehr viel mehr Daten aufkommen und die Anlage auch weitaus mehr Nebenläufigkeit aufweist als die PPU, wurde hier versucht mit wenigen Werkstück-Durchläufen das Verhalten vollständig zu beobachten, um einen möglichst überschaubaren Datensatz zu generieren, der es erlaubt die Ergebnisse der Fallstudie auch interpretieren zu können. Daher wurden zur Durchführung der Fallstudie so viele Werkstücke bearbeitet, dass jede Bearbeitungsreihenfolge drei- bis viermal beobachtet werden konnte. Dies war ausreichend, um die Modelle aus den Daten zu generieren. Es sei erwähnt, dass die Identifikation der Werkstücke mittels Zählen der Anzahl an Auslösungen der Reedkontakte den Annahmen des MFPN-Lernalgorithmus widerspricht. Gemäß Annahme 9-2 löst jede Werkstückinstanz beim Durchlauf durch eine Ressource ein Signal maximal einmal aus. Bei den Werkstücktypen, die einen Reedkontakt zweimal auslösen, wird diese Annahme verletzt. Aus diesem Grund musste dieser Zusammenhang bei der Ausführung des Algorithmus manuell nachgetragen werden, indem den entsprechenden Ereignissen manuell die richtige ID zugewiesen wurde. Tabelle 11-5 fasst die Rahmenbedingungen für die Durchführung der Fallstudie zusammen.

**Tabelle 11-5: Rahmenbedingungen bei der Durchführung der Fallstudie für die Evolutionsszenarien**

Evolutionsszenario		Sc1	Sc2	Sc3
<b>Rahmenbedingung</b>				
<b>#Werkstücke</b>	<b>rot</b>	1	1	1
	<b>blau</b>	1	1	1
	<b>grün</b>	1	1	1
	<b>gelb</b>	1	1	3
	<b>natur</b>	3	3	4
<b># Signale</b>		110	125	125
<b># Ereignisse</b>		1020	1784	2690
<b>Parameter</b>	<b><i>T<sub>thresh</sub></i></b>	60	60	60
	<b><i>cost</i></b>	5000	5000	5000
	<b><i>maxDistance</i></b>	10000	10000	10000
	<b><i>maxOverlap</i></b>	300	300	300

<sup>54</sup> Das vollständige Informationsmodell ist im Anhang in Tabelle F-2 gegeben.

### 11.2.3 Diskussion der Ergebnisse

Zunächst sei erwähnt, dass für alle betrachteten Szenarien MSPNs und je ein MFPN generiert werden konnten. Eine Sichtprüfung der Modelle ergab, dass sie das Verhalten (abzüglich einer Toleranz in den Zeitwerten) korrekt wiedergeben. Tabelle 11-6 gibt die Eckdaten der Petri-Netze an. Repräsentativ wurde hierbei für jede „Art“ technischer Ressource ein MSPN aus den insgesamt 21 (Szenario 1) bzw. 25 (Szenarien 2 und 3)<sup>55</sup> MSPNs ausgewählt. Diese sind Drehtisch 4.1 als Repräsentant für alle Drehtische, Förderband 4.2 als Repräsentant für alle langen Förderbänder mit zwei Näherungssensoren, Förderband 4.3 als Repräsentant für alle kurzen Förderbänder mit einem Näherungssensor und Maschine M2 als Repräsentant für alle Werkzeugmaschinen. Aus den gegebenen Eckdaten lassen sich die gleichen Erkenntnisse gewinnen, wie aus den Eckdaten der für die PPU generierten Petri-Netze.

Eine Analyse der Modelle wurde vorgenommen<sup>56</sup>. In erster Linie zeigen die Ergebnisse, dass das Ändern des Routings zur Flexibilisierung der Prozesse einen Anstieg der Produktionsdauer je Werkstücktyp zur Folge hat. Dies drücken die ermittelten mittleren Produktionsdauern aus. Somit zeigt sich, dass die Maßnahmen, mehrere Werkstücke in einer Teilanlage zuzulassen und die Bearbeitungsdauern an den Maschinen M2 und M3 zu verringern, nicht ausreichen, um die längeren Transportwege zu kompensieren.

Tabelle 11-6: Eckdaten der generierten Petri-Netze

		Sc1	Sc2	Sc3
<i>MSPN<sup>Table4.1</sup></i>	$ \mathcal{T} $	18	21	26
	$ \mathcal{S}_{mach} $	14	12	15
	$ \mathcal{S}_{sig} $	14	12	12
<i>MSPN<sup>Conv4.2</sup></i>	$ \mathcal{T} $	8	6	9
	$ \mathcal{S}_{mach} $	6	5	6
	$ \mathcal{S}_{sig} $	6	6	6
<i>MSPN<sup>Conv4.3</sup></i>	$ \mathcal{T} $	5	6	4
	$ \mathcal{S}_{mach} $	3	4	3
	$ \mathcal{S}_{sig} $	4	4	4
<i>MSPN<sup>M2</sup></i>	$ \mathcal{T} $	18	15	15
	$ \mathcal{S}_{mach} $	15	11	12
	$ \mathcal{S}_{sig} $	18	18	18
<i>MFPN</i>	$ \mathcal{T} $	26	48	64
	$ \mathcal{S} $	26	47	63

Im Gegensatz zur PPU konnte hier kein Zusammenhang zwischen Werkstückpositionen und Bearbeitung durch Maschinen mittels gleicher Zugehörigkeit entsprechender Signale im Informationsmodell ermittelt werden, da die Werkzeugmaschinen keinen Werkstück detektierenden Sensor enthalten. Vielmehr musste dieser Zusammenhang anhand der Korrelation der 1-Zustände der Werkstück bearbeitenden Signale und der Werkstück detektierenden Signale ermittelt werden (siehe hierzu auch das Beispiel zu Beginn von Kapitel 10). Hiermit konnte für jede Werkzeugmaschine in jedem Szenario die zugehörige Position im Materialfluss korrekt und eindeutig ermittelt werden. Somit konnten auch für alle Szenarien die Menge an benötigten Operationen je Werkstücktyp sowie die Prozessfolgen ermittelt werden. Für naturfarbene Werkstücke in Sc3 wurden allerdings eine falsche Prozessfolge und somit auch ein falscher

<sup>55</sup>Es sei erwähnt, dass in Szenario 1 ein Drehtisch und drei Förderbänder nicht genutzt werden

<sup>56</sup>Die Analyse ergab die im Anhang in den Tabellen F-3 – F-5 dargelegten Merkmalsausprägungen

Operationsvorrat ermittelt. Naturfarbene Werkstücke werden in diesem Szenario zwar nicht bearbeitet, jedoch müssen sie, um an Anlagenausgang 2 aussortiert zu werden, Maschine M3 passieren. Da die Analyse des MFPN nur ermittelt, ob sich Werkstücke an bzw. in der Maschine befunden haben, nicht jedoch, ob sie dann auch bearbeitet wurden, wird hier fälschlicherweise eine Bearbeitung angenommen. Es kann jedoch automatisch ermittelt werden, dass der Werkstück bearbeitende Aktor in M3 seltener angesteuert wurde (6x), als Werkstücke an der Maschine waren (10 Stück). Welche Werkstücke bearbeitet wurden, kann jedoch nicht automatisiert ermittelt werden. Daher ist eine Interpretation der Analyseergebnisse durch Anlagenpersonal mit entsprechendem Know-how notwendig.

Wie in der PPU-Fallstudie hat sich gezeigt, dass der Ansatz anwendbar ist und zu nachvollziehbaren Ergebnissen führt, die sich zur Bewertung der Evolution eignen. Jedoch ist auch hier eine korrekte Interpretation der Ergebnisse durch anlagenkundiges Personal vorzunehmen, um die richtigen Schlussfolgerungen aus den Ergebnissen ziehen zu können.

## 12 Bewertung, Zusammenfassung und Ausblick

### 12.1 Bewertung und Grenzen des Ansatzes

Das vorangegangene Kapitel hat die grundlegende Anwendbarkeit des in dieser Arbeit vorgestellten Ansatzes aufzeigen können. Dieser Abschnitt soll nun den Ansatz kritisch betrachten und bewerten. Dies wird einerseits anhand der Evaluierungsergebnisse geschehen, andererseits auch anhand theoretischer Überlegungen und Eigenschaften der vorgestellten Methode und den zugehörigen Algorithmen. Dabei soll der Ansatz insbesondere hinsichtlich der in Kapitel 6 dargelegten Anforderungen und der daraus resultierenden These diskutiert werden. Die folgenden Absätze beziehen sich daher auf die genannten Anforderungen.

#### **Anforderung 6-1: Anlagenevolution muss erkannt und bewertet werden**

Der vorliegende Ansatz beruht auf der Generierung von Verhaltensmodellen auf Basis von Beobachtungen des Anlagenverhaltens. Hierzu wurden zwei Arten von Modellen definiert. MSPN erlauben die Darstellung der vollständigen Sprache der technischen Ressourcen in Form von Feuersequenzen der Petri-Netze. Sie enthalten nicht nur die kausalen Zusammenhänge der Signalverläufe, sondern auch die zeitlichen Zusammenhänge. In den Fallstudien konnten für alle betrachteten technischen Ressourcen korrekte MSPN-Modelle generiert werden. Ein aus der Ereignissequenz generiertes MFPN zeigt zusätzlich den Materialfluss in der Fertigungsanlage auf. In Kapitel 10 wurden verschiedene Algorithmen aufgezeigt, die sich zur Ermittlung von Messelementen eignen. Diese können zur Bestimmung verschiedener Qualitäts-Merkmalausprägungen herangezogen werden. Dass diese sich zur Bewertung der Anlagenevolution eignen, wurde in den Fallstudien aufgezeigt (siehe hierzu auch untenstehende Diskussion zu Anforderung 6-4).

In Analogie zur Domäne der Fehlererkennung können MSPN und MFPN als aus Daten generierte Modelle zur Erkennung abnormalen Verhaltens herangezogen werden. Hierzu wurden in den Abschnitten 8.5 und 9.4 Anomalien definiert, die dem Verhalten der generierten Modelle widersprechen. Es ist jedoch nicht garantiert, dass es sich bei erkannten Anomalien um eine Anlagenevolution handelt. Eine erkannte Anomalie gibt lediglich einen Hinweis auf ein beobachtetes Verhalten, das nicht vom Modell abgedeckt wird. Wie allgemein bei Verfahren, die datenbasierte Modellgenerierung nutzen, unterliegt auch dieser Ansatz der Prämisse, dass das Verhalten vollständig und fehlerfrei beobachtet wurde, um ein vollständiges und korrektes Modell generieren zu können. Für die Fallstudien konnte vollständiges Verhalten recht überschaubar und mittels der Bearbeitung weniger Werkstücke hervorgerufen werden. Anlagen im industriellen Einsatz weisen in der Regel jedoch einen weitaus höheren Komplexitätsgrad auf. Häufig kann gar nicht eingeschätzt werden, ob und wann das Verhalten vollständig beobachtet wurde, oder es ist schlichtweg nicht möglich das Verhalten vollständig herbei zu führen. Grundsätzlich ist es jedoch auch möglich, Modelle aus unvollständigen Verhaltensbeobachtungen zu generieren. Allerdings ist in diesem Fall eine Anomalie stets darauf zu prüfen, ob sie aus neuem Verhalten oder lediglich bisher nicht beobachtetem Verhalten resultiert.

Zudem kann auch nicht garantiert werden, dass die Modelle das beobachtete Verhalten und insbesondere alle kausalen Zusammenhänge gänzlich beinhalten. Insbesondere wenn Abhängigkeiten zwischen einzelnen technischen Ressourcen vorherrschen, die nicht durch den Materialfluss bedingt sind, werden diese nicht in den Modellen abgebildet, da für die technischen Ressourcen separate Modelle generiert werden. Dass für jede technische Ressource je ein MSPN generiert wird, hat jedoch den Vorteil, dass die

MSPN überschaubar bleiben, wie auch an den gegebenen Eckdaten der generierten Modelle für die Fallstudien zu erkennen ist. Häufig arbeiten die Ressourcen unabhängig voneinander und parallel. Die Trennung der MSPN bewirkt, dass daraus resultierende Nebenläufigkeiten nicht explizit abgebildet werden (müssen). Die Trennung der Modelle nach technischen Ressourcen entspricht somit einem Kompromiss zwischen reduzierter Modellkomplexität und vollständiger Abdeckung des beobachteten Verhaltens. Dies muss bei der Interpretation der Modelle und auch der Anomalien berücksichtigt werden.

#### **Anforderung 6-2: Manueller Aufwand sollte möglichst gering gehalten werden**

Die einzige aktive manuelle Beteiligung sieht der Ansatz in der Erstellung des Informationsmodells vor. Für die beiden Fallstudienanlagen konnten Informationsmodelle in kurzer Zeit und ohne besonderen Aufwand erstellt werden. Hierzu war lediglich die Kenntnis über die Zugehörigkeit von Sensoren und Aktoren zu technischen Ressourcen und die Bedeutung der zugehörigen Signale notwendig. Dieses Wissen ist üblicherweise vorhanden und ist bspw. auch zur Erstellung von Signallisten während des Engineerings der Fertigungsanlage notwendig und kann diesen später auch entnommen werden. Insbesondere ist es vorteilhaft, dass zur Erstellung des Informationsmodells kein besonderes methodisches Wissen oder Kenntnisse in der Modellierung notwendig sind. Vielmehr kann das Informationsmodell in einfacher tabellarischer Darstellung und unter Hinzunahme der bereits vordefinierten Signaltypen (siehe Abschnitt 7.3) vorgenommen werden, wie auch anhand der Fallstudien aufgezeigt. Hier wurden die Informationsmodelle in einfachen 2-dimensionalen Zell-Array-Strukturen abgelegt und von den Algorithmen verwendet.

Es sei jedoch zu beachten, dass industrielle Anlagen häufig eine vielfach höhere Anzahl an I/O-Signalen aufweisen als die betrachteten Fallstudien. Der manuelle Aufwand zur Erstellung des Informationsmodells skaliert linear mit der Menge an Signalen. Somit kann die Erstellung des Informationsmodells in der industriellen Praxis entsprechend aufwändiger ausfallen. Es kann dennoch allgemeingültig angenommen werden, dass die manuelle Erstellung von Verhaltensmodellen und deren Analyse stets einen ungleich höheren Aufwand erfordern. Somit liefert der Ansatz einen Gewinn bzgl. der Reduktion von manuellen Aufwänden.

#### **Anforderung 6-3: Dynamische Informationen sollten extern verfügbar sein**

Diese Anforderung deckt der Ansatz vollständig ab, da er lediglich auf I/O-Signale zugreift. Somit ist keine Kenntnis über den Quellcode der SPS vonnöten und auch die Ausführung der Steuerung wird durch den Ansatz nicht beeinflusst. Es sei hier allerdings erwähnt, dass der Zugriff auf weitere Informationen den Ansatz durchaus sinnvoll ergänzen könnte. Die Kenntnis über bestimmte interne SPS-Zustände könnte helfen die Modelle besser zu interpretieren oder weitere Modellarten zu definieren, die eine Analyse hinsichtlich zusätzlicher Merkmalsausprägungen erlauben. Gleiches gilt z.B. auch, wenn der Zugriff zu weiteren Automatisierungssystemen wie SCADA- oder MES-Systemen oder deren Schnittstellen betrachtet würde.

#### **Anforderung 6-4: Die Analyse muss auf einem hohen Abstraktionsniveau vorgenommen werden**

Die Bewertung des Anlagenverhaltens wird mittels Ermittlung von Merkmalsausprägungen ermöglicht, die für die Berechnung typisch relevanter nicht-funktionaler Eigenschaften herangezogen werden können. Welche Eigenschaften sich hierfür eignen, wurde anhand einer ausführlichen Literaturrecherche ermittelt (siehe Abschnitt 3.3). Die Fallstudien haben gezeigt, dass eine Vielzahl an Zeit-, Mengen- und Strukturgrößen aus den generierten Modellen ermittelt werden konnten. Diese können sogar mithilfe des MFPN auf einzelne Werkstücktypen bezogen werden, was eine differenzierte Bewertung des Anlagenverhalten möglich macht. Es hat sich jedoch auch gezeigt, dass unter Umständen fehlerbehaftete

Werte oder lediglich Abschätzungen der Werte ermittelt werden können. Zum Beispiel können Prozessschritte nicht eindeutig identifiziert werden. Diese werden abgeschätzt mittels der ausführenden Maschine für einen Prozessschritt. Somit sind Analyseergebnisse ebenfalls stets durch einen Anlagenkundigen zu interpretieren.

Zudem liegt ein Nachteil des Ansatzes in dessen reaktivem Charakter. Ein Modell kann erst erlernt werden, wenn das zugehörige Verhalten bereits beobachtet wurde. Somit kann Verhalten der Anlage auch erst bewertet werden, wenn es bereits umgesetzt wurde. Da eine a priori-Bewertung in der Praxis jedoch häufig vernachlässigt oder gar nicht durchgeführt wird (vgl. Abschnitt 5.2.3), liefert der Ansatz dennoch eine gewinnbringende Unterstützung.

Zusammenfassend lässt sich aussagen, dass der vorgestellte Ansatz hilft, Anlagenevolution auf hohem Abstraktionsniveau zu bewerten. Jedoch bleibt die Herausforderung für den Anwender des Ansatzes, die Ergebnisse der Algorithmen zu interpretieren und zu bewerten. Dies kann nur durch entsprechend anlagenkundiges Personal vorgenommen werden. Fazit der Untersuchungen ist, dass sich die in Kapitel 6 formulierte These durch die dargelegten Forschungsergebnisse bestätigen lässt. Der manuelle Aufwand zur Analyse und Bewertung konnte gering gehalten werden, insbesondere im Vergleich zu dem Aufwand, der bei der manuellen Erstellung von Verhaltensmodellen nötig wäre. Gleichzeitig konnten Analyseergebnisse generiert werden, die ein hohes Abstraktionsniveau aufweisen und Merkmalsausprägungen von Eigenschaften entsprechen, die üblicherweise für Anlagenbetreiber relevant und ihnen entsprechend vertraut sind.

## 12.2 Zusammenfassung

Produktionssysteme stehen heutzutage einem immer dynamischeren Umfeld gegenüber. Volatile Beschaffungs- und Absatzmärkte, ein steigender Bedarf an individualisierten Produkten sowie sich ändernde Umwelteinflüsse und Technologien gehören zu den wichtigsten Gründen für sich stetig ändernde Anforderungen an die zu produzierenden Produkte und somit auch an das Produktionssystem. Daraus resultierende vorgenommene Änderungen an der Anlagenphysik sowie der Steuerungssoftware aber auch Alterungserscheinungen wie Verschleiß der Anlage wurden im Rahmen der vorliegenden Arbeit als *Anlagenevolution* zusammengefasst. Produktionssysteme unterliegen üblicherweise einer stetigen Evolution während ihrer gesamten Betriebsphase. Meist weisen industrielle Produktionssysteme einen hohen Komplexitätsgrad auf, weshalb die vollständigen Auswirkungen, auch kleinerer Änderungen, häufig kaum abschätzbar sind. Von besonderer Bedeutung ist daher, dass die Änderungen zum Einen gut dokumentiert und zum Anderen bewertet werden. Nur so kann sichergestellt werden, dass Anlagenwissen über evolvierende Produktionssysteme explizit vorhanden bleibt und die Auswirkungen der Änderungen bekannt und gewollt oder bewusst tolerierbar sind.

Zwei Arten von wissenschaftlichen Ansätzen wurden hinsichtlich ihrer Eignung zur Unterstützung von Anlagenevolution untersucht. Modellbasierte Ansätze verwenden Struktur- und/oder Verhaltensmodelle zum Engineering und zur Analyse von Systemen oder Teildisziplinen der Systeme (z.B. Automatisierungstechnik). Wesentlicher Vorteil solcher Ansätze ist der hohe Abstraktionsgrad der verwendeten Modelle. Meist sind die Modelle mit einem bestimmten Ziel, wie der Analyse hinsichtlich bestimmter Merkmale, dem Testen oder der Quellcodegenerierung für die Automatisierung erstellt worden und spiegeln die hierfür relevanten Eigenschaften verständlich wieder. Sie sind entsprechend für den Menschen leicht interpretierbar und dienen als gutes Dokumentationsmaterial. Dem gegenüber steht allerdings der hohe Aufwand, der zur Erstellung der Modelle notwendig ist. Die Modellerstellung muss von einem Ingenieur vorgenommen werden, der nicht nur die zu modellierende Anlage und die Produkti-

onsprozesse hinreichend gut kennt, sondern auch die Modellierungssprache. Häufig sind gewisse Parameter nicht bekannt und können nur angenommen oder geschätzt werden. Dies, sowie der hohe Zeit- und Kostendruck bei der Durchführung von Änderungen, sind die wesentlichen Gründe, weshalb in der industriellen Praxis häufig keine detaillierten Modelle vorhanden sind oder Modelle nicht aktuell gehalten werden. Entsprechend bleibt auch eine Bewertung von Änderungen aus. Die zweite Art untersuchter Methoden sind modelllernende Ansätze wie sie bspw. aus der Fehlerdiagnose oder dem Process Mining bekannt sind. Sie erlauben eine nahezu aufwandslose Modellerstellung, da sie auf Beobachtungen des Systemverhaltens basieren. Während des Betriebes generierte Daten werden herangezogen, um die Modelle mittels Lernalgorithmen automatisch zu generieren. Problematisch bei dieser Art von Ansätzen ist jedoch, dass die resultierenden Modelle einen sehr geringen Abstraktionsgrad aufweisen. Sie bieten nicht die Möglichkeit abstrakte Kennwerte zu ermitteln, die typischerweise für die Bewertung von Produktionssystemen herangezogen werden.

Ziel des hier vorgestellten Ansatzes war es, eine evolutionsunterstützende Methode aufzuzeigen, die insbesondere der Bewertung von Änderungen anhand von Ausprägungen anerkannter Qualitätsmerkmale dient. Hierzu kombiniert der Ansatz modellbasierte und modelllernende Verfahren. Der manuelle Aufwand zur Durchführung des Ansatzes sollte dabei möglichst gering gehalten werden, um o.g. Defiziten entgegenzuwirken. Der allgemeine Ansatz sieht vor, die dynamischen Daten, die an der Schnittstelle zwischen dem technischen Prozess und der Anlagenautomation aufkommen, zu verwenden, um automatisiert Verhaltensmodelle der Anlage zu generieren. Um das Abstraktionsniveau der generierten Modelle und damit deren Interpretierbarkeit hinsichtlich der abgezielten Eigenschaften anzuheben, werden statische a-priori Informationen hinzugezogen. Sie werden in Form eines Informationsmodells entsprechenden Algorithmen zur Generierung und zur Analyse der Verhaltensmodelle zur Verfügung gestellt. Da die Erstellung des Informationsmodells manuell erfolgen muss, sollte eine möglichst geringe Menge an Informationen verwendet werden. Daher enthält das Informationsmodell lediglich einfache Semantiken zu den Signalen, die für die Modellgenerierung herangezogen werden. Im Speziellen sind dies die Zugehörigkeit der Signale zum Equipment bzw. zu den technischen Ressourcen der Anlage sowie deren Bedeutung, also welche Information ein Signal enthält.

Zur Realisierung des o.g. Ansatzes in Form konkreter Algorithmen wurden im Rahmen dieser Arbeit speziell Fertigungssysteme als Teilmenge aller Produktionssysteme betrachtet. Fertigungssysteme zeichnen sich dadurch aus, dass sie der Produktion diskreter Güter dienen. Zudem beinhalten sie überwiegend bis vollständig binäre Sensorik und Aktorik. Folglich lassen sie sich als *Discrete Event Systems* beschreiben. Die betrachteten Systeme wurden in dieser Arbeit weiter eingegrenzt, indem sie sich als eine Menge von Maschinen zur Werkstückbearbeitung (z.B. Werkzeugmaschinen) sowie Transport- und Materialflussequipment (z.B. Förderbänder) beschreiben lassen. Es wurde ein Satz an einfachen Signaltypen für diese Klasse von Fertigungssystemen zur Erstellung eines Informationsmodelles entwickelt. Des Weiteren wurden zwei Arten von Petri-Netz-Modellen mitsamt zugehöriger Lern- und Analysealgorithmen vorgestellt. *Machine State Petri Nets* dienen der Darstellung des kausalen und zeitbehafteten Verhaltens einzelner technischer Ressourcen. Sie bilden das Verhalten detailliert ab. Dadurch eignen sie sich zur Anomalieerkennung im Verhalten der einzelnen Ressourcen sowie für die Dokumentation des Verhaltens. *Material Flow Petri Nets* bilden den Materialfluss innerhalb des Fertigungssystems ab. Neben den Transportwegen beinhalten sie die Transportzeiten einzelner Abschnitte, die sich zudem anhand der beförderten Werkstücktypen unterscheiden lassen, um eine differenzierte Analyse zu ermöglichen. Kern des Lernalgorithmus für diese Materialflussmodelle ist die auf einer Clusteranalyse basierende Ermittlung zeitstabiler Abstände zwischen gleichen Ereignissen.

Es wurden verschiedene Analysealgorithmen aufgezeigt, die auf o.g. Modelle angewandt werden können, um relevante Eigenschaften von Fertigungssystemen zu bestimmen. Aufgrund ihrer Allgemeingültigkeit wurde hier der Fokus auf nicht-funktionale Eigenschaften gelegt. Welche Eigenschaften im Bereich der industriellen Fertigung bekannt und relevant sind, wurde anhand einer umfassenden Literaturrecherche ermittelt. Vorwiegend aus einschlägigen Normen wurden Metriken zur Bestimmung der Merkmalsausprägungen dieser Eigenschaften recherchiert. Hieraus ergab sich eine Menge an Messelementen, die zur Berechnung der Merkmalsausprägungen herangezogen werden können. Ein Großteil dieser kann mithilfe der generierten Modelle ermittelt werden. Dies konnte auch anhand zweier Fallstudien aufgezeigt werden. Hierbei handelt es sich um eine Pick-and-Place Unit sowie eine Job-Shop-Fertigung im Labormaßstab. Die Ergebnisse der Fallstudien haben die Umsetzbarkeit des Ansatzes und seiner Algorithmen aufgezeigt. Es zeigte sich jedoch auch, dass Modelle und Analyseergebnisse einer Interpretation durch anlagenkundiges Personal bedürfen, um die Ergebnisse korrekt einschätzen zu können.

### **12.3 Weiterer Handlungsbedarf**

Allgemein existieren drei verschiedene Handlungsfelder, die diese Arbeit fortführenden Forschungsbedarf aufzeigen. Hierbei handelt es sich um die Übertragung des Ansatzes auf weitere Systemklassen, die Erforschung neuer und Optimierung der vorhandenen Algorithmen sowie die Erarbeitung von Ansätzen, die das extrahierte Wissen in Form von Modellen und Analyseergebnissen sinnvoll weiter verwenden. Diese drei Handlungsfelder werden im Folgenden genauer erläutert:

#### **Übertragung des Ansatzes auf weitere Systemklassen**

Die Algorithmen, die im Rahmen dieser Arbeit vorgestellt wurden, beschränken die betrachteten Systeme auf Fertigungssysteme, die als Discrete Event Systems modellierbar sind. Inwieweit sich der Ansatz in Form der in Abschnitt 7.1 aufgezeigten allgemeinen Wissensarchitektur eignet, um weitere Systemklassen zu betrachten, konnte in dieser Arbeit nicht betrachtet werden. In der Domäne der Produktionsautomatisierung ist insbesondere von Interesse, ob und wie sich der Ansatz für verfahrenstechnische Anlagen eignet. Hierzu ist es notwendig, die Systeme als kontinuierliche oder hybride Systeme zu betrachten, vgl. [NiFR15]. Auch Systeme anderer Domänen, z.B. informationstechnische Systeme, könnten von der Methode profitieren. Allgemein könnten alle langlebigen und Evolution unterliegenden Systeme als Betrachtungsgegenstand dienen.

Je nach betrachteter Systemklasse sind auch andere Merkmale zur Bewertung der Systeme zu berücksichtigen. Entsprechend sind andere Modelle und Beschreibungsmittel sowie auch Lern- und Analysealgorithmen vonnöten. Unter der Annahme, dass die Wissensarchitektur generisch genug ist, müssen hierzu insbesondere Algorithmen und Methoden zur Modellgenerierung und –Analyse gefunden werden, die sich auf die betrachtete Systemklasse anwenden lassen. Die vorliegende Arbeit hat bereits aufgezeigt, wie dabei vorgegangen werden kann. Zunächst sollte ermittelt werden, welche Eigenschaften für die betrachtete Systemklasse relevant und geeignet sind, um zur Bewertung von geändertem Verhalten herangezogen werden zu können. Des Weiteren müssen diese Eigenschaften in operationalisierter Form vorliegen oder operationalisiert werden. Das heißt, sie müssen schlussendlich in Form von Metriken vorliegen, die mithilfe vorhandener Daten bzw. daraus generierten Modellen berechnet werden können. Im Hinblick auf diese Metriken müssen Modelle definiert und Lernalgorithmen entworfen oder adaptiert werden. Beispiele für bestehende Algorithmen existieren für hybride [NiG14] und kontinuierliche [Ise06] Produktionssysteme. Schlussendlich müssen Anomalien definiert werden und Analysealgorithmen zur Bestimmung der Eigenschaften gefunden werden.

### Optimierung der vorgestellten Algorithmen und Einbringen neuer Algorithmen

Um das gesamte Potential der Nutzung der Signalsemantiken zur Modellgenerierung und –Analyse zu ergründen, bedarf es der Weiterentwicklung der hier vorgestellten Algorithmen sowie der Erforschung weiterer Algorithmen. Einige Anregungen hierzu werden in den folgenden Absätzen gegeben:

Abhängigkeiten zwischen den gelernten Modellen könnten explizit gelernt werden, um die Anomalieerkennung und Analysen zu verbessern. Ein möglicher Ansatz zur Ermittlung von Abhängigkeiten zwischen nebenläufigen, aus Daten generierten Modellen findet sich bspw. in [RLL10].

Zudem gehen die bisherigen Lernalgorithmen stets davon aus, dass ein vollständiger Datensatz vorliegt, aus dem die Modelle gelernt werden können. Die Modelle müssen somit stets von Grund auf neu generiert werden. Es ist keine iterative Adaption der gelernten Modelle möglich, die es erlaubt, neu entdecktes Verhalten alten Modellen hinzuzufügen. Eine Methode, die dies ermöglichen könnte, ist die Online Identifikation, vgl. [DFM07, DFM+08].

Für die Ermittlung der Zeitannotationen könnten Kerndichteschätzer verwendet werden, wie in [FoVo13] zur Ermittlung von Ausführungszeiten einzelner Fertigungsaufgaben vorgeschlagen. Vorteil hierbei ist, dass ein Kerndichteschätzer es erlaubt, Verteilungen mit mehreren Maxima zu erlernen. Somit können Zeiten, die situationsbedingt unterschiedlich sein können, getrennt voneinander betrachtet werden. Eine Adaption der hier vorgestellten Algorithmen könnte unterschiedliche Petri-Netz-Pfade für unterschiedliche Zeitverteilungen generieren, so dass unterschiedliche Situationen auch in der Struktur der Netze erkennbar sind.

Um den reaktiven Charakter des Ansatzes zu mindern, ist die Kombination mit modellgetriebenen Ansätzen denkbar. Erste Ideen und Ansätze dazu wurden in [HLL+14c<sup>#</sup>] zur Kombination des hier vorgestellten Ansatzes mit modellgetriebenen Engineering-Methoden und [LHL+15<sup>#</sup>] zur Kombination mit modellgetriebenem Testen aufgezeigt.

Des Weiteren benötigen die Algorithmen zur Modellgenerierung derzeit eine manuelle Parametrierung. Dies betrifft insbesondere die Zeitdifferenz zur Kombination von Ereignissen  $T_{thresh}$  (vgl. Abschnitt 8.2) sowie den Schwellwert  $cost$  zur Wahl geeigneter Cluster, die stabile Zeitdifferenzen enthalten, im Rahmen der MFPN-Generierung (vgl. Abschnitt 9.2.3). Des Weiteren wurden zur effizienten und robusten Umsetzung des Algorithmus zur MFPN-Generierung die Parameter  $maxDistance$  und  $maxOverlap$  eingeführt (vgl. Abschnitt 11.1.2). Die Parametrierung der Algorithmen bedarf zusätzlichem manuellen Aufwand sowie spezifischem Anlagewissen. Dies reduziert die Erfüllung der Forderung nach geringem manuellen Aufwand und möglichst wenig Wissen über das Anlagenverhalten. Eine Möglichkeit dem entgegenzuwirken, könnte ein heuristisches Verfahren zur automatisierten Ermittlung möglichst optimaler Parameter der Algorithmen sein. Ein auf den Ereignissequenzen angewandtes analytisches Verfahren ist ebenfalls denkbar. Die Untersuchung verschiedener Verfahren zur möglichst optimalen Parametrierung der Algorithmen sei daher als weiterer Forschungsbedarf genannt.

Zudem können weitere Potentiale des generellen Ansatzes ausgeschöpft werden, wenn dem Informationsmodell weitere oder andere Daten hinzugefügt werden. Zudem ist die Nutzung weiterer Informationsquellen wie SCADA- oder MES-Systemen denkbar.

### Weiterverwendung extrahierten Wissens

Die hier dargestellten Forschungsergebnisse wurden überwiegend im Forschungsprojekt *Forever Young Production Automation with Active Components (FYPA<sup>2</sup>C)* im Rahmen des Schwerpunktprogrammes *SPP1593 Design For Future – Managed Software Evolution* der deutschen Forschungsgesellschaft (siehe

[DFG17<sup>@</sup>] erzielt. In der zweiten Förderperiode dieses Schwerpunktprogrammes findet seit September 2016 eine Fortführung des Forschungsprojektes statt. In dem Projekt wird untersucht, inwieweit sich das mithilfe des hier vorgestellten Ansatzes ermittelbare Wissen über vergangene Evolutionsschritte einer Anlage für zukünftige Evolution nutzen lässt. Hierzu soll ein Netzwerk aus evolvierenden Systemen betrachtet werden. Dieses macht es möglich, evolutionsrelevantes Wissen für ein System aus den Erfahrungen anderer Systeme zu generieren. Evolutionsrelevantes Wissen soll dabei mittels Modelldeltas beschrieben werden, die die Änderung der generierten Modelle vor und nach einem Evolutionsschritt beschreiben. Diese werden mit den resultierenden Änderungen der Merkmalsausprägungen verknüpft und entsprechend zur Auswertung für künftige Evolution anderer Systeme zur Verfügung gestellt. Zu beachten ist hierbei, dass Produktionssysteme unterschiedlichen Kontexteinflüssen unterliegen und auch die Systeme selbst stets Unterschiede aufweisen. Daher werden Metriken erforscht, die es erlauben Ähnlichkeiten und Unterschiede in den Systemmodellen und im Systemkontext zu bewerten.

## Anhang A: Verwendete Formelzeichen und Notation

Mengen (im mathematischen Sinne) werden stets als fett gedruckte Großbuchstaben dargestellt. Elemente innerhalb von Mengen werden mit demselben, nicht fett gedruckten Großbuchstaben und einem entsprechenden Index dargestellt. Der Index entspricht demselben Buchstaben, jedoch klein gedruckt. Dies gilt sowohl für lateinische als auch für griechische Buchstaben. Die Anzahl aller in einer Menge befindlichen Elemente werden mittels  $|\cdot|$ , also der Kardinalität der Menge, dargestellt.

### Beispiel:

Die Menge aller Werkstücke bearbeitenden Maschinen ist  $\mathbf{M}$ . Die Elemente der Menge werden entsprechend mit  $M_m$  bezeichnet und die Anzahl aller Maschinen im System ist als  $|\mathbf{M}|$  notiert. Demnach gilt auch  $0 \leq m \leq |\mathbf{M}|$ . Die Menge aller Operationen wird mit  $\mathbf{\Omega}$ , ihre Elemente mit  $\Omega_\omega$  und die Anzahl aller Operationen mit  $|\mathbf{\Omega}|$  notiert.

Entsprechend wird in der folgenden Tabelle, wenn nicht weiter erforderlich, für Mengengrößen lediglich die Notation der Menge angegeben, da sich der entsprechende Index und die Bezeichnung der Elemente daraus ergeben.

**Tabelle A-1: Liste der verwendeten Formelzeichen**

$A$	Verfügbarkeit (Availability) – Mit Index $r$ bezogen auf eine einzelne technische Ressource. Mit Index $w$ bezogen auf die Produktion des Produkttypen $w$
$\mathcal{A}$	Zustandsautomat
$Annot(\mathcal{T}_t)$	Menge aller an Transition $\mathcal{T}_t$ annotierten Elemente
$AR_w$	Anzahl Routen, die Werkstücktyp durch das Fertigungssystem nehmen kann
$\alpha_{mw}$	Operative Fähigkeit der Maschine $m$ bezogen auf Produkttyp $w$
$a_{wi}$	Anteil gemeinsam nutzbarer Operationen für die Fertigung von Produkten des Typen $w$ und $i$
$AM_w$	Ausschussmenge des Werkstücktypen $w$ : Anzahl aufgrund minderwertiger Qualität aussortierter Werkstücke. Ohne Index angegeben, aggregiert über alle Werkstücktypen.
$AQ_w$	Ausschussquote des Werkstücktypen $w$ . Ohne Index angegebenen Index aggregiert über alle Werkstücktypen.
$B(E^r)$	Ereignispropagationsmatrix der Ereignissequenz $E^r$
$b_{uv}$	Element der Ereignispropagationsmatrix
$BLG_r$	Beleggrad der technischen Ressource $r$ . Alternativ kann auch der Index $m$ für den Beleggrad der Maschine $m$ verwendet werden.
$\mathcal{C}$	Inzidenzmatrix eines Petri-Netzes
$\mathcal{C}^+$	Eingangsinzidenzmatrix
$\mathcal{C}^-$	Ausgangsinzidenzmatrix
$c_{ik}$	Element der $i$ -ten Zeile und $k$ -ten Spalte der Inzidenzmatrix
$C_w$	Produktionskapazität für Werkstücktyp $w$ , falls auf einzelne Maschinen bezogen, mit hochgestelltem Index $m$
$\Delta$	Sensitivitätsmaß zur Erkennung nicht mehr vorhandenen Verhaltens
$d_{max}$	Maximale Aktivierungsdauer einer Transition / Maximale Aufenthaltsdauer an detektierter Position (MFPN)

$d_{min}$	Minimale Aktivierungsdauer einer Transition (MSPN) / Minimale Aufenthaltsdauer an de- tektierter Position (MFPN)
$DS_w$	Durchsatz des Werkstücktypen $w$ . Ohne Index angegeben, aggregiert über alle Werkstück- typen
$\tilde{e}(\mathcal{T}_t)$	An Transition $\mathcal{T}_t$ annotiertes (kombiniertes) Ereignis
$e_j(k)$	Ereignis an der Stelle $j$ der Ereignissequenz zum Zeitpunkt $k \cdot T_{OPC}$
$e_j^r(k)$	Ereignis an der Stelle $j$ der Ereignissequenz, das im Informationsmodell technischer Res- source $r$ zugewiesen ist zum Zeitpunkt $k \cdot T_{OPC}$
$\tilde{e}_j^r(k)$	Kombiniertes Ereignis an der Stelle $j$ der Ereignissequenz, das im Informationsmodell technischer Ressource $r$ zugewiesen ist zum Zeitpunkt $k \cdot T_{OPC}$
$e_{\omega w}^m$	Effizienz, mit der Maschine $m$ Operation $\omega$ zur Bearbeitung von Werkstücktypen $w$ ausfüh- ren kann (Dauer der Operation). Ohne Index $w$ , kein Bezug auf Werkstücktypen.
$E_{Mat}^{glob}$	Materialfluss relevante Ereignissequenz zur Zuordnung globaler IDs; Generiert, enthält Eintritts- und Austrittssequenzen
$E_{Mat}^j$	Materialfluss relevante Ereignissequenz, die lediglich Ereignisse mit zugewiesener ID $j$ enthält
$E_{Mat}^r$	Ereignissequenz der technischen Ressource $r$ , die ausschließlich Materialfluss relevante Signale enthält; ohne hochgestelltes $r$ , über alle Ressourcen hinweg
$E_{obs}^r$	Beobachtete Ereignissequenz der technischen Ressource $r$ ; ohne hochgestelltes $r$ , über alle Ressourcen hinweg
$Eff$	Effektivität, Anlagenweit (mit Index $m$ angegeben: Effektivität einer einzelnen Maschine)
$\mathcal{F}$	Menge der Flussrelationen eines Petri-Netzes
$\mathcal{F}_{mach}$	Flussrelationen entsprechend der Kanten zwischen Transitionen und auf Maschinenver- halten bezogenen Stellen eines MSPN
$\mathcal{F}_{sig}$	Flussrelationen entsprechend der Kanten zwischen Transitionen und auf Signalverhalten bezogenen Stellen eines MSPN
$FPY_w$	First Pass Yield des Werkstücktypen $w$ . Ohne Index bezogen auf die gesamte Produkti- onsmenge
$\mathcal{G}$	Menge der Zustandsübergangsfunktionen (eines Automaten)
$g_\omega$	Gewichtung der Fertigungsoperation $\omega$ (Häufigkeit der Operation im Verhältnis zur Ge- samtanzahl durchgeführter Operationen)
$GM_w$	Gutmenge des Werkstücktyps $w$ : Anzahl fehlerfrei produzierter (bzw. nicht aufgrund min- derer Qualität aussortierter) Produkte
$GT_w$	Anzahl an Gutteilen des Werkstücktypen $w$ . Gutteile sind diejenigen, die im ersten Prüfdurchgang als qualitativ gut befunden wurden.
$I_\nu$	Einheitsmatrix der Größe $\nu \times \nu$
<b>ident</b>	Vektor aus Elementarzuständen von Werkstück identifizierenden Signalen
$K_{mw\omega}$	Fähigkeitsindex der Maschine $m$ bezogen auf Operation $\omega$ zur Produktion von Produkttyp $w$
$\lambda_r$	Ausfallrate der technischen Ressource $r$ .
<b>M</b>	Menge aller Werkstück bearbeitenden Maschinen (z.B. Werkzeugmaschinen, Montagero-

	boter)
$m$	Markierung eines Petri-Netzes, mit Index 0: Anfangsmarkierung
$\mu$	Mittlere Aktivierungsdauer einer Transition (MSPN) / Mittlere Aufenthaltsdauer an detek- tierter Position (MFPN)
$M_{\Omega_w}$	Menge der Maschinen, die Operation $\omega$ zur Bearbeitung von Werkstücktyp $w$ durchführen können
$MHF$	Material Handling Flexibility. Mit Index $w$ auf einzelnen Werkstücktypen bezogen.
$MF_m$	Machine Flexibility der Maschine $m$
$MFPN_w$	Teil-MFPN, das lediglich den Materialfluss des Werkstücktypen $w$ beschreibt
$MSPN$	Machine State Petri Net
MTBF	Mean Time Between Failure, mittlere Zeit zwischen zwei Ausfällen eines Systems. Mit Index $r$ bezogen auf einzelne technische Ressource
MTTR	Mean Time To Repair mittlere Zeit zur Wiederaufnahme des Betriebs eines Systems. Mit Index $r$ bezogen auf einzelne technische Ressource
$N_w$	Anzahl an Prozessfolgen (Arbeitsplänen) zur Fertigung von Produkttyp $w$
$n$	Anzahl an Feuervorgängen einer Transition während eines Beobachtungszeitraumes (MSPN) / Anzahl an Transportvorgängen an detektierter Position (MFPN)
$n'(T_t, k)$	Anzahl an Feuervorgängen der Transition $T_t$ während der Anomalieerkennung zum Zeit- punkt $k$
$v$	Anzahl unterschiedlicher Ereignisse in Ereignissequenz $E^r$ definiert.
$NG_m$	Nutzgrad der Maschine $m$
$NM_w$	Nacharbeitsmenge des Werkstücktypen $w$ : Menge der Werkstücke, die im Laufe der Pro- duktion nachgearbeitet werden mussten. Ohne Index angegeben, aggregiert über alle Werkstücktypen.
$NQ_w$	Nacharbeitsquote des Werkstücktypen $w$ . Ohne Index angegeben, aggregiert über alle Werkstücktypen
$OF_w$	Operation Flexibility zur Fertigung von Produkttyp $w$
$\Omega$	Menge aller disjunkten Fertigungsoperationen
$\Omega^m$	Menge der durch Maschine $m$ durchführbaren Operationen
$\Omega^w$	Menge aller Operationen, die zur Herstellung von Produkten des Typs $w$ benötigt werden
$\Omega_{\omega}^m$	Operation $\omega$ , die an Maschine $m$ ausgeführt wird/werden kann
$\Omega_{\omega}^w$	Operation $\omega$ , die zur Bearbeitung von Werkstücktyp $w$ benötigt wird
$\Omega_{\omega}^{mw}$	Bearbeitung von Werkstücktyp $w$ an Maschine $m$ durch Operation $\omega$
$OEE$	Gesamtanlageneffektivität (Overall Equipment Effectiveness)
$P_m$	Produktivität der Maschine $m$ . Ohne Index aggregiert für gesamtes Produktionssystem
$p_{mw}$	Wahrscheinlichkeit der Zuordnung von Produkttyp $w$ zu Maschine $m$
$p_w$	Wahrscheinlichkeit der Produktion von Produkttypen $w$ , bzw. Anteil an Menge produzier- ter Produkte des Typen $w$ an Gesamtproduktionsmenge
$PF$	Process Flexibility

$PG$	Prozessgrad
$\mathcal{PN}$	Petri-Netz
$PM_w$	Produzierte Menge des Werkstücktypen $w$ , bezogen auf die Gesamtproduktionsdauer
$PR_w$	Produktionsrate für Werkstücke des Typen $w$
$PT_w$	Anzahl an geprüften Teilen des Werkstücktypen $w$
$QR_w$	Qualitätsrate zu Produkttyp $w$ . Ohne Index: Aggregiert über alle Produkttypen
$R$	Menge aller technischen Ressourcen einer Fertigungsanlage
$R(S_i)$	Technische Ressource, der im Informationsmodell Signal $S_i$ zugewiesen wurde
$RF_w$	Routing Flexibility bezogen auf Werkstück Typ $w$
$\mathcal{S}$	Menge der Stellen eines Petri-Netzes
$\bullet \mathcal{S}_s$	Vorbereich der Stelle $\mathcal{S}_s$
$\mathcal{S}_s \bullet$	Nachbereich der Stelle $\mathcal{S}_s$
$\mathcal{S}$	Menge der I/O-Signale
$s_i^\uparrow(t)$	Steigende Flanke des $i$ -ten Signals zum Zeitpunkt $t$
$s_i^\downarrow(t)$	Fallende Flanke des $i$ -ten Signals zum Zeitpunkt $t$
$s_i^{high}$	1-Zustand des $i$ -ten Signals
$s_i^{low}$	0-Zustand des $i$ -ten Signals
$\mathcal{S}_{mach}$	Menge an Stellen eines MSPN mit Bezug zum Maschinenverhalten
$\mathcal{S}_{sig}$	Menge an Stellen eines MSPN mit Bezug zum Signalverhalten
$\sigma$	Standardabweichung der Aktivierungsdauer einer Transition (MSPN) / Standardabweichung der Aufenthaltsdauer an detektierter Position (MFPN)
$\mathcal{ST}$	Menge der Signaltypen
$ST(S_i)$	Im Informationsmodell Signal $i$ zugewiesener Signaltyp
$\mathcal{T}$	Menge an Transport- und Materialflussequipment
$\mathcal{J}$	Menge der Transitionen eines Petri-Netzes
$T_{OPC}$	OPC Serverzykluszeit
$T_{thresh}$	Schwelldauer, innerhalb derer Ereignisse zu einem kombinierten Ereignis zusammengefasst werden
$\bullet \mathcal{J}_t$	Vorbereich der Transition $\mathcal{J}_t$
$\mathcal{J}_t \bullet$	Nachbereich der Transition $\mathcal{J}_t$
$t_p$	Gesamtproduktionsdauer (resp. Dauer des Beobachtungszeitraumes)
$t_m^{HN}$	Hauptnutzungszeit der Maschine $m$ , entspricht der Zeit in der die Maschine wertschöpfende Aktivitäten durchführt
$t_m^{SU}$	Kumulierte Dauer der störungsbedingten Unterbrechungen an Maschine $m$
$t_r^B$	Belegungszeit der technischen Ressource $r$ , approximiert durch die Zeit, die die technische Ressource nicht im Ruhezustand ist. Alternativ kann auch der Index $m$ für die Belegungszeit der Maschine $m$ verwendet werden. Bezogen auf die Gesamtproduktionsdauer

---

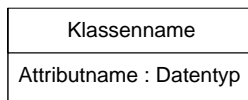
$t_w^\mu$	Mittlere Produktionsdauer für Werkstücke des Typen $w$ (Zykluszeit)
$t_w^{max}$	Maximale Produktionsdauer für Werkstücke des Typen $w$
$t_w^{min}$	Minimale Produktionsdauer für Werkstücke des Typen $w$
$U_w$	Auslastung bezogen auf Produkttyp $w$
$v$	Vektor, der die T-Invariante eines Petri-Netzes beschreibt
$W$	Menge der Werkstücktypen
$x_{ij}$	Binäre Variable zur Angabe, ob Transportverbindung zwischen zwei Maschinen $i$ und $j$ existiert. Mit hochgestelltem Index $w$ bezogen auf bestimmten Werkstücktypen
$Z$	Zuverlässigkeit. Mit Index $r$ bezogen auf eine einzelne technische Ressource. Mit Index $w$ bezogen auf die Produktion des Produkttypen $w$
$\mathcal{Z}$	Menge der Zustände (eines Zustandsautomaten)
$Z(x)$	Elementarzustand des Signals $x$
$z_0$	Initialzustand (eines Zustandsautomaten)

## Anhang B: Notation der Unified Modeling Language

Die vorliegende Arbeit verwendet zur Darstellung einiger Konzepte und Abhängigkeiten Diagramme der Unified Modeling Language (UML). Die in dieser Arbeit genutzte Notation der Elemente wird im Folgenden kurz erläutert. Für weitergehende Informationen sei bspw. auf [OEs09] verwiesen.

### Klassendiagramm

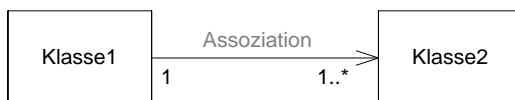
#### Klasse



**Abbildung B -1: Klasse**

Eine Klasse wird durch ein Rechteck dargestellt. Der Text im obersten Bereich des Rechtecks entspricht dem Klassennamen. Besitzt eine Klasse gewisse Attribute, so sind diese im unteren Teil der Klasse beschrieben. Das Attribut besteht aus einem Attributnamen gefolgt von einem Doppelpunkt und dem Datentypen des Attributs. Enthalten die Klassen keine Attribute, so kann der Attribute-Bereich auch entfallen bzw. wird nicht dargestellt. Klassen beschreiben allgemeine abstrakte Objekttypen. Instanzen der Klassen werden Objekte genannt. Die Beziehungen, die Objekte und/oder Klassen zueinander haben können, werden mittels *Assoziationen* dargestellt.

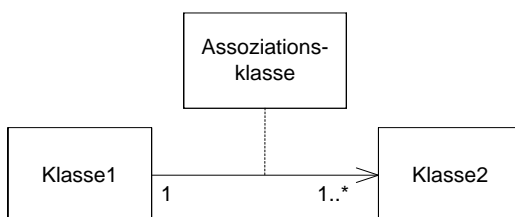
#### Assoziation



**Abbildung B-2: Assoziation**

Eine Assoziation ist durch eine Verbindung zwischen Klassen dargestellt. Sie beschreibt allgemein eine Beziehung zwischen den verbundenen Klassen. Eine Assoziation kann ungerichtet oder gerichtet sein. Im Falle einer gerichteten Assoziation ist die Richtung mittels einer Pfeilspitze dargestellt (am obigen Beispiel von Klasse1 zu Klasse2). Die Bedeutung der Assoziation kann als Text an die Assoziation geschrieben sein. Sollte eine Assoziation zwischen mehreren Objekten der verbundenen Klassen bestehen können, so wird dies mithilfe von *Multiplizitäten* an den Assoziationsenden dargestellt. Eine natürliche Zahl gibt dabei die genaue Anzahl an. Ein möglicher Bereich ist durch „..“ angegeben (z.B. 1..3). Ist die Anzahl beliebig (inkl. 0), so wird dies mit einem „\*“ angegeben. Dieses Symbol kann auch genutzt werden, um ein beliebiges Ende eines Intervalls anzugeben (z.B. 1..\*).

#### Assoziationsklassen



**Abbildung B -3: Assoziationsklasse**

Entstammt eine Assoziation einer ebenfalls modellierten Klasse und soll dies explizit angegeben werden, so kann dies mithilfe einer gestrichelten Linie zwischen der Assoziationsklasse und der Assoziation ge-

schehen. In UML gibt es einige vordefinierte spezielle Assoziationen, von denen die hier verwendeten untenstehend mit deren spezifischer Notation erläutert sind.

### Generalisierung



Abbildung B-4: Generalisierung

Die Generalisierung zeigt eine Vererbungshierarchie zwischen zwei Klassen an. Die erbende Klasse (Subklasse in der Abbildung) erbt dabei alle Eigenschaften der vererbenden Klasse (Klasse), kann aber weitere Eigenschaften, die sie weiter spezialisieren, aufweisen.

### Aggregation



Abbildung B-5: Aggregation

Die Aggregation gibt eine „Teil von“ Beziehung zwischen Klassen oder Objekten an. Sie beschreibt, wie sich etwas Ganzes aus Teilen zusammensetzt.

### Komposition



Abbildung B-6 Komposition

Die Komposition ist eine strenge Form der Aggregation. Hier sind die Teile vom ganzen existenzabhängig, d.h. sie können ohne das Ganze nicht existieren. Eine beliebig tiefe Hierarchie kann mit einer Komposition von einer Klasse auf sich selbst dargestellt werden:

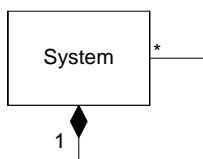


Abbildung B-7: Systemhierarchie

## Anhang C: Zusammenfassende Darstellung der Messelemente für Qualitätsmerkmale

Die Vielzahl an diskutierten Metriken (Abschnitt 3.3) zeigt, wie vielfältig die Bewertungskriterien eines Fertigungssystems zusammengestellt werden können. Allerdings zeigt sich auch, dass einige Merkmalsausprägungen als Basis-Messungen zur Ermittlung der Ausprägung mehrerer (zusammengesetzter) Merkmale genutzt werden können. Folgende Tabelle fasst die Merkmale als Basis-Messelemente zusammen und zeigt auf, zu welchen Qualitätsmerkmalen diese beitragen.

Tabelle C-1: Basis-Messungen zur Ermittlung der NFEs

Merkmalklasse	Merkmalssträger	Messelement	Benötigt zur Bestimmung von
Zeiten und zeitbezogene Größen	Maschine	Belegungszeit, $t_r^B$ :	Beleggrad (3-1)
		Effizienz/Dauer der Durchführung von Operationen, $e_{\omega w}^m$	Machine Flexibility (3-22)(3-26)
		Hauptnutzungszeit der Maschinen, $t_m^{HN}$	Nutzgrad (3-4) Effektivität (3-5) Techn. Nutzgrad (3-8), Prozessgrad (3-9) Produktivität (3-16)
		Belegungszeit der Maschinen, $t_m^B$	Nutzgrad (3-4)
		Kumulierte Dauer störungsbedingter Unterbrechungen, $t_m^{SU}$	Techn. Nutzgrad (3-8)
	Gesamtsystem/ Produkttyp	Minimale Produktionsdauer, $t_w^{min}$	Effektivität (3-5) Kapazität (3-13) Auslastung (3-15)
		Mittlere Produktionsdauer, $t_w^{\mu}$	Durchsatz (3-3) Beleggrad (3-1) Effektivität (3-5) Qualitätsrate (3-6) Auslastung (3-15)
		Maximale Produktionsdauer, $t_w^{max}$	Effektivität (3-5)
	Gesamtsystem	Gesamtproduktionsdauer, $t_p$	Beleggrad (3-1) Durchsatz (3-2) Prozessgrad (3-9) Auslastung (3-14) Produktivität (3-16)
	(Produkt-) Mengen	Gesamtsystem/ Produkttyp	Produzierte Menge, $PM_w$

Merkmal-Klasse	Merkmalsträger	Messelement	Benötigt zur Bestimmung von
			Auslastung (3-14) Machine Flexibility (3-25)
		Gutmenge, $GM_w$	Qualitätsrate (3-6)
		Anzahl an Gutteilen, $GT_w$	First Pass Yield (3-10)
		Anzahl Prüfteile $PT_w$	First Pass Yield (3-10)
		Ausschussmenge, $AM_w$	Ausschussquote (3-11)
		Nacharbeitsmenge $NM_w$	Nacharbeitsquote (3-12)
Topologie- und Strukturgrößen	Transportsystem	Transportverbindungen zwischen Maschinen, $x_{ij}$	Verfügbarkeit (3-18)(3-19) Zuverlässigkeit (3-20), Mat. Handling Flex. (3-27)
		Anzahl (Transport-)Routen je Werkstück $AR_w$	Routing Flexibility (3-28)
	Gesamtsystem	Anzahl Maschinen $ M $	Mat. Handling Flex. (3-27) Process Flexibility (3-30) Produktivität (3-16)
		Anordnung technischer Ressourcen (seriell/parallel)	Verfügbarkeit (3-18)(3-19) Zuverlässigkeit (3-20)
Planung und Fähigkeiten	Produkt	Prozessfolgen	Verfügbarkeit (3-18)(3-19) Zuverlässigkeit (3-20) Operation Flexibility (3-29)
		Für Produktfertigung benötigte Operationen $\Omega^w$	Machine Flexibility (3-24) Operation Flexibility (3-29) Process Flexibility (3-31)
	Maschine(n)	Durchführbare Operationen (je Maschine und Gesamt)	Machine Flexibility (3-21) Machine Flexibility (3-24)
	Gesamtsystem	Häufigkeit (Gewichtung) der Ausführung von Operationen	Machine Flexibility (3-22)(3-26)
		Anzahl Produkttypen $ W $	Process Flexibility (3-30)
		Häufigkeit der Zuordnung Maschine/Produkttyp $p_{mw}$	Process Flexibility (3-30)
Ausfälle	Technische Ressource	Mean Time Between Failure, $MTBF_r$	Verfügbarkeit (3-17)
		Mean Time To Repair, $MTTR_r$	Verfügbarkeit (3-17)
		Ausfalldichte, $\lambda_r$	Zuverlässigkeit (3-20)

## Anhang D: Ereignissequenzen der Beispiele

Tabelle D-1: Vollständige Ereignissequenz zum in Abschnitt 8.4 dargestellten Beispiel

Zeitstempel in ms	Signal	Ereignis
388400	'B3_A12'	1
389976	'B3_S09'	1
389992	'B3_A12'	0
398088	'B3_A12'	1
398244	'B3_S09'	0
399430	'B3_A12'	0
435091	'B3_A12'	1
436698	'B3_A12'	0
436698	'B3_S09'	1
444795	'B3_A12'	1
444951	'B3_S09'	0
446183	'B3_A12'	0
604476	'B3_A12'	1
606083	'B3_S09'	1
606099	'B3_A12'	0
614226	'B3_A12'	1
614382	'B3_S09'	0
615599	'B3_A12'	0
681322	'B3_A12'	1
682898	'B3_A12'	0
682898	'B3_S09'	1
691088	'B3_A12'	1
691244	'B3_S09'	0
692492	'B3_A12'	0
782254	'B3_A12'	1
783783	'B3_S09'	1
783799	'B3_A12'	0
792051	'B3_A12'	1
792192	'B3_S09'	0
793440	'B3_A12'	0

Tabelle D-2: Vollständige Ereignissequenz zum in Abschnitt 9.2.3 gezeigten Beispiel

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis	ID
1	83974	'B1_S02'	1	1
2	87562	'B1_S02'	0	1
3	89278	'B1_S03'	1	1
4	90682	'B1_S03'	0	1
5	92242	'B1_S07'	1	1
6	93022	'B1_S07'	0	1
7	94895	'B1_S08'	1	1
8	95051	'B1_S08'	0	1
9	99107	'B1_S09'	1	1
10	99887	'B1_S09'	0	1
11	101759	'B2_S06'	1	1
12	102227	'B2_S06'	0	1
13	104255	'B2_S09'	1	1
14	104411	'B2_S09'	0	1
15	107375	'B2_S10'	1	1
16	107843	'B2_S10'	0	1
17	123911	'B1_S02'	1	2

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis	ID
18	127499	'B1_S02'	0	2
19	129215	'B1_S03'	1	2
20	130463	'B1_S03'	0	2
21	132023	'B1_S07'	1	2
22	132803	'B1_S07'	0	2
23	134519	'B1_S08'	1	2
24	134675	'B1_S08'	0	2
25	138887	'B1_S09'	1	2
26	139667	'B1_S09'	0	2
27	141539	'B2_S06'	1	2
28	141851	'B2_S06'	0	2
29	143879	'B2_S09'	1	2
30	144035	'B2_S09'	0	2
31	146999	'B2_S10'	1	2
32	147467	'B2_S10'	0	2
33	167279	'B2_S03'	1	4
34	168839	'B2_S03'	0	4
35	169775	'B1_S02'	1	3
36	170711	'B2_S08'	1	4
37	170867	'B2_S08'	0	4
38	172271	'B2_S07'	1	4
39	173363	'B1_S02'	0	3
40	174143	'B2_S07'	0	4
41	175079	'B1_S03'	1	3
42	175859	'B2_S06'	1	4
43	176327	'B1_S03'	0	3
44	177575	'B2_S06'	0	4
45	177887	'B1_S07'	1	3
46	178667	'B1_S07'	0	3
47	179603	'B2_S09'	1	4
48	179759	'B2_S09'	0	4
49	180383	'B1_S08'	1	3
50	180539	'B1_S08'	0	3
51	182723	'B2_S10'	1	4
52	183191	'B2_S10'	0	4
53	184751	'B1_S09'	1	3
54	185531	'B1_S09'	0	3
55	187403	'B2_S06'	1	3
56	187715	'B2_S06'	0	3
57	189743	'B2_S09'	1	3
58	190055	'B2_S09'	0	3
59	193019	'B2_S10'	1	3
60	193487	'B2_S10'	0	3
61	224999	'B2_S03'	1	5
62	226559	'B2_S03'	0	5
63	228431	'B2_S08'	1	5
64	228743	'B2_S08'	0	5
65	229991	'B2_S07'	1	5
66	231863	'B2_S07'	0	5
67	233579	'B2_S06'	1	5
68	234515	'B2_S03'	1	6
69	235295	'B2_S06'	0	5
70	236075	'B2_S03'	0	6
71	237479	'B2_S09'	1	5
72	237635	'B2_S09'	0	5
73	237947	'B2_S08'	1	6
74	238259	'B2_S08'	0	6
75	239507	'B2_S07'	1	6
76	240599	'B2_S10'	1	5

Lfd. Nr.	Zeitstempel in ms	Signal	Ereignis	ID
77	242315	'B2_S07'	0	6
78	242471	'B2_S10'	0	5
79	244031	'B2_S06'	1	6
80	245747	'B2_S06'	0	6
81	247775	'B2_S09'	1	6
82	247931	'B2_S09'	0	6
83	250895	'B2_S10'	1	6
84	252767	'B2_S10'	0	6

Tabelle D-3: Ereignissequenz  $E_{Mat}^{glob}$  zur Zuweisung globaler IDs zum in Abschnitt 9.2.4 gezeigten Beispiel

Zeitstempel in ms	Signal	Ereignis	Ressource	Interne ID	Globale ID
1921	WorkpieceReady'	1	2	1	1
2881	CranSucked&CranOnMagazin&CranDown'	1	1	1	1
2881	WorkpieceReady'	0	2	1	1
5701	MagazinCapacitiveSensor'	1	2	2	2
6181	CranSucked&CranOnConveyor&CranDown'	0	1	1	1
9181	CranSucked&CranOnMagazin&CranDown'	1	1	2	2
9241	WorkpieceReady'	0	2	2	2
12241	WorkpieceReady'	1	2	3	3
14342	StampSliderFilled'	1	3	1	2
14402	CranSucked&CranPositionStamp&CranDown'	0	1	2	2
17042	CranSucked&CranPositionStamp&CranDown'	1	1	7	2
17222	StampSliderFilled'	0	3	1	2
20401	CranSucked&CranOnConveyor&CranDown'	0	1	7	2
23404	CranSucked&CranOnMagazin&CranDown'	1	1	3	3
23404	WorkpieceReady'	0	2	3	3
26477	WorkpieceReady'	1	2	4	4
26658	CranSucked&CranOnConveyor&CranDown'	0	1	3	3
29724	CranSucked&CranOnMagazin&CranDown'	1	1	4	4
29724	WorkpieceReady'	0	2	4	4
32548	MagazinCapacitiveSensor'	1	2	5	5
32972	CranSucked&CranOnConveyor&CranDown'	0	1	4	4
36035	CranSucked&CranOnMagazin&CranDown'	1	1	5	5
36097	WorkpieceReady'	0	2	5	5
39105	WorkpieceReady'	1	2	6	6
41145	StampSliderFilled'	1	3	2	5
41265	CranSucked&CranPositionStamp&CranDown'	0	1	5	5
43848	CranSucked&CranPositionStamp&CranDown'	1	1	8	5
44030	StampSliderFilled'	0	3	2	5
47222	CranSucked&CranOnConveyor&CranDown'	0	1	8	5
50228	CranSucked&CranOnMagazin&CranDown'	1	1	6	6
50228	WorkpieceReady'	0	2	6	6
53532	CranSucked&CranOnConveyor&CranDown'	0	1	6	6

Tabelle D-4: Ereignissequenz  $E_{Mat}$  mit zugewiesenen globalen IDs zum in Abschnitt 9.3 gezeigten Beispiel

Zeitstempel in ms	Signal	Ereignis	Globale ID
1921	'WorkpieceReady'	1	1
2881	'CranSucked&CranOnMagazin&CranDown'	1	1
2881	'WorkpieceReady'	0	1
3001	'CranSucked&CranOnMagazin&CranDown'	0	1
3181	'CranSucked&CranOnMagazin&CranUp'	1	1
3301	'CranSucked&CranOnMagazin&CranUp'	0	1
5161	'CranSucked&CranOnConveyor&CranUp'	1	1
5461	'CranSucked&CranOnConveyor&CranUp'	0	1
5701	'MagazinCapacitiveSensor'	1	2
5941	'WorkpieceReady'	1	2
6061	'CranSucked&CranOnConveyor&CranDown'	1	1
6181	'CranSucked&CranOnConveyor&CranDown'	0	1
9181	'CranSucked&CranOnMagazin&CranDown'	1	2
9241	'MagazinCapacitiveSensor'	0	2
9241	'WorkpieceReady'	0	2
9301	'CranSucked&CranOnMagazin&CranDown'	0	2
9481	'CranSucked&CranOnMagazin&CranUp'	1	2
9601	'CranSucked&CranOnMagazin&CranUp'	0	2
11461	'CranSucked&CranOnConveyor&CranUp'	1	2
11581	'CranSucked&CranOnConveyor&CranUp'	0	2
12241	'WorkpieceReady'	1	3
13442	'CranSucked&CranPositionStamp&CranUp'	1	2
13682	'CranSucked&CranPositionStamp&CranUp'	0	2
14162	'CranSucked&CranPositionStamp&CranDown'	1	2
14342	'StampSliderFilled'	1	2
14402	'CranSucked&CranPositionStamp&CranDown'	0	2
17042	'CranSucked&CranPositionStamp&CranDown'	1	2
17162	'CranSucked&CranPositionStamp&CranDown'	0	2
17222	'StampSliderFilled'	0	2
17342	'CranSucked&CranPositionStamp&CranUp'	1	2
17462	'CranSucked&CranPositionStamp&CranUp'	0	2
19380	'CranSucked&CranOnConveyor&CranUp'	1	2
19680	'CranSucked&CranOnConveyor&CranUp'	0	2
20280	'CranSucked&CranOnConveyor&CranDown'	1	2
20401	'CranSucked&CranOnConveyor&CranDown'	0	2
23404	'CranSucked&CranOnMagazin&CranDown'	1	3
23404	'WorkpieceReady'	0	3
23532	'CranSucked&CranOnMagazin&CranDown'	0	3
23652	'CranSucked&CranOnMagazin&CranUp'	1	3
23772	'CranSucked&CranOnMagazin&CranUp'	0	3
25697	'CranSucked&CranOnConveyor&CranUp'	1	3
25937	'CranSucked&CranOnConveyor&CranUp'	0	3
26477	'WorkpieceReady'	1	4
26597	'CranSucked&CranOnConveyor&CranDown'	1	3
26658	'CranSucked&CranOnConveyor&CranDown'	0	3
29724	'CranSucked&CranOnMagazin&CranDown'	1	4
29724	'WorkpieceReady'	0	4
29846	'CranSucked&CranOnMagazin&CranDown'	0	4
29966	'CranSucked&CranOnMagazin&CranUp'	1	4
30147	'CranSucked&CranOnMagazin&CranUp'	0	4
32007	'CranSucked&CranOnConveyor&CranUp'	1	4
32249	'CranSucked&CranOnConveyor&CranUp'	0	4
32548	'MagazinCapacitiveSensor'	1	5
32791	'WorkpieceReady'	1	5

Zeitstempel in ms	Signal	Ereignis	Globale ID
32911	CranSucked&CranOnConveyor&CranDown'	1	4
32972	CranSucked&CranOnConveyor&CranDown'	0	4
36035	CranSucked&CranOnMagazin&CranDown'	1	5
36097	MagazinCapacitiveSensor'	0	5
36097	WorkpieceReady'	0	5
36157	CranSucked&CranOnMagazin&CranDown'	0	5
36277	CranSucked&CranOnMagazin&CranUp'	1	5
36458	CranSucked&CranOnMagazin&CranUp'	0	5
38323	CranSucked&CranOnConveyor&CranUp'	1	5
38683	CranSucked&CranOnConveyor&CranUp'	0	5
39105	WorkpieceReady'	1	6
40245	CranSucked&CranPositionStamp&CranUp'	1	5
40545	CranSucked&CranPositionStamp&CranUp'	0	5
41025	CranSucked&CranPositionStamp&CranDown'	1	5
41145	StampSliderFilled'	1	5
41265	CranSucked&CranPositionStamp&CranDown'	0	5
43848	CranSucked&CranPositionStamp&CranDown'	1	5
43968	CranSucked&CranPositionStamp&CranDown'	0	5
44030	StampSliderFilled'	0	5
44150	CranSucked&CranPositionStamp&CranUp'	1	5
44281	CranSucked&CranPositionStamp&CranUp'	0	5
46201	CranSucked&CranOnConveyor&CranUp'	1	5
46501	CranSucked&CranOnConveyor&CranUp'	0	5
47101	CranSucked&CranOnConveyor&CranDown'	1	5
47222	CranSucked&CranOnConveyor&CranDown'	0	5
50228	CranSucked&CranOnMagazin&CranDown'	1	6
50228	WorkpieceReady'	0	6
50348	CranSucked&CranOnMagazin&CranDown'	0	6
50468	CranSucked&CranOnMagazin&CranUp'	1	6
50648	CranSucked&CranOnMagazin&CranUp'	0	6
52508	CranSucked&CranOnConveyor&CranUp'	1	6
52751	CranSucked&CranOnConveyor&CranUp'	0	6
53412	CranSucked&CranOnConveyor&CranDown'	1	6
53532	CranSucked&CranOnConveyor&CranDown'	0	6

Tabelle D-5: Ereignissequenz zur Generierung des MSPN einer Werkzeugmaschine zum in Abschnitt 10.1 gezeigten Beispiel mit Signaltypen

Zeitstempel in ms	Signal	Ereignis	Signaltyp
66347	'B3_A08'	1	'A.Move'
66503	'B3_A05'	1	'A.Move'
66503	'B3_A08'	0	'A.Move'
66503	'B3_S05'	0	'S.State'
68063	'B3_A05'	0	'A.Move'
68063	'B3_S04'	1	'S.State'
68219	'B3_A07'	1	'A.Move'
68219	'B3_A09'	1	'A.Modify'
68219	'B3_S07'	0	'S.State'
69935	'B3_A07'	0	'A.Move'
69935	'B3_S06'	1	'S.State'
72899	'B3_A08'	1	'A.Move'
73211	'B3_S06'	0	'S.State'
76175	'B3_S07'	1	'S.State'

<b>Zeitstempel in ms</b>	<b>Signal</b>	<b>Ereignis</b>	<b>Signaltyp</b>
76331	'B3_A06'	1	'A.Move'
76331	'B3_A08'	0	'A.Move'
76331	'B3_A09'	0	'A.Modify'
76331	'B3_S04'	0	'S.State'
77891	'B3_A06'	0	'A.Move'
77891	'B3_S05'	1	'S.State'
203159	'B3_A05'	1	'A.Move'
203315	'B3_S05'	0	'S.State'
204875	'B3_A05'	0	'A.Move'
204875	'B3_A07'	1	'A.Move'
204875	'B3_A09'	1	'A.Modify'
204875	'B3_S04'	1	'S.State'
204875	'B3_S07'	0	'S.State'
206591	'B3_A07'	0	'A.Move'
206591	'B3_S06'	1	'S.State'
209711	'B3_A08'	1	'A.Move'
209867	'B3_S06'	0	'S.State'
212987	'B3_A06'	1	'A.Move'
212987	'B3_A08'	0	'A.Move'
212987	'B3_A09'	0	'A.Modify'
212987	'B3_S07'	1	'S.State'
213143	'B3_S04'	0	'S.State'
214547	'B3_A06'	0	'A.Move'
214547	'B3_S05'	1	'S.State'
423743	'B3_A05'	1	'A.Move'
423743	'B3_S05'	0	'S.State'
425459	'B3_A05'	0	'A.Move'
425459	'B3_A07'	1	'A.Move'
425459	'B3_A09'	1	'A.Modify'
425459	'B3_S04'	1	'S.State'
425459	'B3_S07'	0	'S.State'
427175	'B3_A07'	0	'A.Move'
427175	'B3_S06'	1	'S.State'
430295	'B3_A08'	1	'A.Move'
430451	'B3_S06'	0	'S.State'
433571	'B3_A06'	1	'A.Move'
433571	'B3_A08'	0	'A.Move'
433571	'B3_A09'	0	'A.Modify'
433571	'B3_S07'	1	'S.State'
433727	'B3_S04'	0	'S.State'
435131	'B3_A06'	0	'A.Move'
435131	'B3_S05'	1	'S.State'
599483	'B3_A05'	1	'A.Move'
599611	'B3_S05'	0	'S.State'
601171	'B3_A05'	0	'A.Move'
601171	'B3_S04'	1	'S.State'
601327	'B3_A07'	1	'A.Move'
601327	'B3_A09'	1	'A.Modify'
601327	'B3_S07'	0	'S.State'
603043	'B3_A07'	0	'A.Move'
603043	'B3_S06'	1	'S.State'
606007	'B3_A08'	1	'A.Move'
606319	'B3_S06'	0	'S.State'
609283	'B3_S07'	1	'S.State'
609439	'B3_A06'	1	'A.Move'
609439	'B3_A08'	0	'A.Move'
609439	'B3_A09'	0	'A.Modify'

Zeitstempel			
in ms	Signal	Ereignis	Signaltyp
609439	'B3_S04'	0	'S.State'
610999	'B3_A06'	0	'A.Move'
610999	'B3_S05'	1	'S.State'

Tabelle D-6: Ereignissequenz  $E_{Mat}$  zum in Kapitel Abschnitt 10.1 gezeigten Beispiel

Zeitstempel			
in ms	Signal	Ereignis	ID
42635	'B1_S02'	1	1
45287	'B1_S02'	0	1
45599	'B1_S04'	1	1
45599	'B1_S05'	1	1
45911	'B1_S04'	0	1
45911	'B1_S05'	0	1
46223	'B1_S04'	1	1
46223	'B1_S05'	1	1
46535	'B1_S04'	0	1
46535	'B1_S05'	0	1
46847	'B1_S03'	1	1
47159	'B1_S03'	0	1
48719	'B1_S07'	1	1
48875	'B1_S07'	0	1
50747	'B1_S08'	1	1
50903	'B1_S08'	0	1
54959	'B1_S09'	1	1
55271	'B1_S09'	0	1
57143	'B2_S06'	1	1
57455	'B2_S06'	0	1
59483	'B2_S09'	1	1
59639	'B2_S09'	0	1
62603	'B2_S10'	1	1
62915	'B2_S10'	0	1
65099	'B3_S03'	1	1
66503	'B3_S03'	0	1
67907	'B3_S09'	1	1
76487	'B3_S09'	0	1
77735	'B3_S10'	1	1
78047	'B3_S10'	0	1
79607	'B3_S14'	1	1
81167	'B3_S14'	0	1
82571	'B3_S11'	1	1
82883	'B3_S11'	0	1
87563	'B4_S16'	1	1
88031	'B1_S02'	1	3
88967	'B4_S16'	0	1
90371	'B4_S11'	1	1
90527	'B4_S11'	0	1
90683	'B1_S02'	0	3
91931	'B4_S10'	1	1
92399	'B1_S03'	1	3
92555	'B1_S03'	0	3
94115	'B1_S07'	1	3
94427	'B1_S07'	0	3
96143	'B1_S08'	1	3
96299	'B1_S08'	0	3
99263	'B4_S10'	0	1
100355	'B1_S09'	1	3

<b>Zeitstempel</b>			
<b>in ms</b>	<b>Signal</b>	<b>Ereignis</b>	<b>ID</b>
100667	'B1_S09'	0	3
100667	'B4_S03'	1	1
102227	'B4_S03'	0	1
102539	'B2_S06'	1	3
103631	'B4_S08'	1	1
103787	'B4_S08'	0	1
104099	'B2_S06'	0	3
105347	'B4_S09'	1	1
105659	'B2_S07'	1	3
105659	'B4_S09'	0	1
105971	'B2_S07'	0	3
107375	'B2_S08'	1	3
107375	'B5_S03'	1	1
107531	'B2_S08'	0	3
108935	'B5_S03'	0	1
109403	'B2_S03'	1	3
110807	'B5_S10'	1	1
117515	'B5_S10'	0	1
119699	'B5_S09'	1	1
120011	'B5_S09'	0	1
122195	'B5_S13'	1	1
135299	'B1_S02'	1	6
137795	'B1_S02'	0	6
138731	'B1_S04'	1	6
139043	'B1_S04'	0	6
139511	'B1_S03'	1	6
139823	'B1_S03'	0	6
141383	'B1_S07'	1	6
141539	'B1_S07'	0	6
143255	'B1_S08'	1	6
143411	'B1_S08'	0	6
144815	'B5_S13'	0	1
147467	'B1_S09'	1	6
191615	'B2_S03'	0	3
191771	'B1_S09'	0	6
193799	'B2_S06'	1	6
193955	'B2_S06'	0	6
195983	'B2_S09'	1	6
196295	'B2_S09'	0	6
199259	'B2_S10'	1	6
199571	'B2_S10'	0	6
201755	'B3_S03'	1	6
203159	'B3_S03'	0	6
204719	'B3_S09'	1	6
213143	'B3_S09'	0	6
214391	'B3_S10'	1	6
214703	'B3_S10'	0	6
216419	'B3_S14'	1	6
217979	'B3_S14'	0	6
219383	'B3_S11'	1	6
219695	'B3_S11'	0	6
222815	'B4_S16'	1	6
224219	'B4_S16'	0	6
225623	'B4_S11'	1	6
225779	'B4_S11'	0	6
227183	'B4_S10'	1	6
234515	'B4_S10'	0	6
236075	'B4_S03'	1	6
237479	'B4_S03'	0	6

<b>Zeitstempel</b>			
<b>in ms</b>	<b>Signal</b>	<b>Ereignis</b>	<b>ID</b>
238883	'B4_S08'	1	6
239039	'B4_S08'	0	6
240755	'B4_S09'	1	6
240911	'B4_S09'	0	6
242471	'B5_S03'	1	6
243875	'B5_S03'	0	6
245747	'B5_S10'	1	6
252611	'B5_S10'	0	6
254639	'B5_S09'	1	6
254795	'B5_S09'	0	6
259319	'B5_S13'	1	6
273515	'B1_S02'	1	4
276167	'B1_S02'	0	4
277883	'B1_S03'	1	4
278039	'B1_S03'	0	4
279599	'B1_S07'	1	4
279755	'B1_S07'	0	4
281627	'B1_S08'	1	4
281783	'B1_S08'	0	4
283187	'B5_S13'	0	6
285995	'B1_S09'	1	4
286151	'B1_S09'	0	4
288023	'B2_S06'	1	4
289583	'B2_S06'	0	4
291143	'B2_S07'	1	4
291299	'B2_S07'	0	4
292703	'B2_S08'	1	4
292859	'B2_S08'	0	4
294731	'B2_S03'	1	4
329831	'B2_S03'	0	4
399719	'B1_S02'	1	7
402371	'B1_S02'	0	7
402683	'B1_S04'	1	7
402839	'B1_S05'	1	7
402995	'B1_S04'	0	7
402995	'B1_S05'	0	7
403307	'B1_S04'	1	7
403619	'B1_S04'	0	7
404087	'B1_S03'	1	7
404243	'B1_S03'	0	7
405959	'B1_S07'	1	7
406115	'B1_S07'	0	7
407831	'B1_S08'	1	7
407987	'B1_S08'	0	7
412199	'B1_S09'	1	7
412511	'B1_S09'	0	7
414383	'B2_S06'	1	7
414539	'B2_S06'	0	7
416567	'B2_S09'	1	7
416879	'B2_S09'	0	7
419843	'B2_S10'	1	7
420155	'B2_S10'	0	7
422339	'B3_S03'	1	7
423743	'B3_S03'	0	7
425303	'B3_S09'	1	7
433727	'B3_S09'	0	7
434975	'B3_S10'	1	7
435443	'B3_S10'	0	7
436847	'B3_S14'	1	7

<b>Zeitstempel</b>			
<b>in ms</b>	<b>Signal</b>	<b>Ereignis</b>	<b>ID</b>
438251	'B3_S14'	0	7
439655	'B3_S11'	1	7
439967	'B3_S11'	0	7
442931	'B4_S16'	1	7
444335	'B4_S16'	0	7
444647	'B1_S02'	1	5
445739	'B4_S11'	1	7
445895	'B4_S11'	0	7
447299	'B1_S02'	0	5
447299	'B4_S10'	1	7
449015	'B1_S03'	1	5
449171	'B1_S03'	0	5
450887	'B1_S07'	1	5
451043	'B1_S07'	0	5
452759	'B1_S08'	1	5
452915	'B1_S08'	0	5
454631	'B4_S10'	0	7
456191	'B4_S03'	1	7
456971	'B1_S09'	1	5
457283	'B1_S09'	0	5
457751	'B4_S03'	0	7
459155	'B2_S06'	1	5
459155	'B4_S08'	1	7
459311	'B4_S08'	0	7
460715	'B2_S06'	0	5
460871	'B4_S09'	1	7
461183	'B4_S09'	0	7
462743	'B5_S03'	1	7
464303	'B5_S03'	0	7
466175	'B5_S10'	1	7
469295	'B2_S07'	1	5
469451	'B2_S07'	0	5
470855	'B2_S08'	1	5
471011	'B2_S08'	0	5
472883	'B2_S03'	1	5
472883	'B5_S10'	0	7
475223	'B5_S09'	1	7
475535	'B5_S09'	0	7
477875	'B5_S13'	1	7
481151	'B2_S03'	0	5
575688	'B1_S02'	1	2
578184	'B1_S02'	0	2
578710	'B1_S04'	1	2
578866	'B1_S04'	0	2
579178	'B1_S05'	1	2
579490	'B1_S05'	0	2
579958	'B1_S03'	1	2
580114	'B1_S03'	0	2
581830	'B1_S07'	1	2
581986	'B1_S07'	0	2
583702	'B1_S08'	1	2
583858	'B1_S08'	0	2
585262	'B5_S13'	0	7
588070	'B1_S09'	1	2
588382	'B1_S09'	0	2
590254	'B2_S06'	1	2
590566	'B2_S06'	0	2
592438	'B2_S09'	1	2
592750	'B2_S09'	0	2

<b>Zeitstempel in ms</b>	<b>Signal</b>	<b>Ereignis</b>	<b>ID</b>
595714	'B2_S10'	1	2
596026	'B2_S10'	0	2
598210	'B3_S03'	1	2
599611	'B3_S03'	0	2
601015	'B3_S09'	1	2
609595	'B3_S09'	0	2
610687	'B3_S10'	1	2
611155	'B3_S10'	0	2
612715	'B3_S14'	1	2
614119	'B3_S14'	0	2
615523	'B3_S11'	1	2
615835	'B3_S11'	0	2
624259	'B4_S16'	1	2
625663	'B4_S16'	0	2
627223	'B4_S11'	1	2
627379	'B4_S11'	0	2
628627	'B4_S10'	1	2
636115	'B4_S10'	0	2
637675	'B4_S03'	1	2
639079	'B4_S03'	0	2
640483	'B4_S08'	1	2
640639	'B4_S08'	0	2
642199	'B4_S09'	1	2
642511	'B4_S09'	0	2
644071	'B5_S03'	1	2
645631	'B5_S03'	0	2
647347	'B5_S10'	1	2
654211	'B5_S10'	0	2
656083	'B5_S09'	1	2
656395	'B5_S09'	0	2
659827	'B5_S13'	1	2
666535	'B5_S13'	0	2

## Anhang E: Daten und Ergebnisse der Anwendung auf der „Pick and Place Unit“

Tabelle E-1: I/O-Signale der PPU sowie Zugehörigkeit zu den Evolutionsszenarien, vgl. [VLF+14]

Signal	Vorhanden ab Szenario	Erläuterung
Actuator_AnalogCranPressure	Sc3	Druck mit dem die Stempelung durchgeführt wird
Actuator_CranLower	Sc1	Befehl Kran hinunterfahren
Actuator_CranTurnClockwise	Sc1	Befehl Kran im Uhrzeigersinn drehen
Actuator_CranTurnCounterclockwise	Sc1	Befehl Kran gegen den Uhrzeigersinn drehen
Actuator_MagazinGreenLamp	Sc1	Befehl grüne „EIN“-Leuchte für Magazin einschalten
Actuator_MagazinSlider	Sc1	Befehl Schieber am Magazin ausfahren (zum Ausschieben von Werkstücken)
Actuator_MagazinVacuumOff	Sc1	Befehl Vakuum des Kranes zum Aufnehmen von Werkstücken einschalten
Actuator_MagazinVacuumOn	Sc1	Befehl Vakuum des Kranes zum Aufnehmen von Werkstücken ausschalten
Actuator_MagazinWhiteLamp	Sc1	Befehl weiße „BEREIT“-Leuchte für Magazin einschalten
Actuator_PusherY1	Sc10	Befehl vorderer Ausschieber am Förderband ausfahren
Actuator_PusherY2	Sc10	Befehl hinterer Ausschieber am Förderband ausfahren
Actuator_SorterConveyorTowardsStacker	Sc10	Befehl Förderband einschalten
Actuator_SorterGreenLampConveyor	Sc10	Befehl grüne „EIN“-Leuchte für Förderband einschalten
Actuator_StampGreenLamp	Sc3	Befehl grüne „EIN“-Leuchte für Stempelschalten
Actuator_StampPusher	Sc3	Befehl zum stempeln
Actuator_StampSliderMovedIn	Sc3	Befehl Schlitten des Stempels zu Stempel bewegen
Actuator_StampSliderMovedOut	Sc3	Befehl Schlitten des Stempels zur Kranübergabestelle bewegen
Actuator_StampWhiteLamp	Sc3	Befehl weiße „BEREIT“-Leuchte für Stempel einschalten
Sensor_CranDown	Sc1	Kran befindet sich an unterer Position
Sensor_CranOnConveyor	Sc1	Kran befindet sich am Förderband
Sensor_CranOnMagazin	Sc1	Kran befindet sich am Magazin
Sensor_CranPositionStamp	Sc3	Kran befindet sich am Stempel
Sensor_CranSucked	Sc1	Kran hat Werkstück angesaugt
Sensor_CranUp	Sc1	Kran befindet sich an oberer Position
Sensor_MagazinCapacitiveSensor	Sc3	Metallisches Werkstück am Magazin erkannt
Sensor_MagazinEmergencyStop	Sc1	Not-Aus-Schalter Magazin
Sensor_MagazinOpticalSensor	Sc10	Helles Werkstück am Magazin erkannt
Sensor_SliderMovedOut	Sc1	Schieber am Magazin ausgefahren
Sensor_SliderNotMovedOut	Sc1	Schieber am Magazin nicht ausgefahren
Sensor_SorterCapacitiveSensorPusher1	Sc11	Metallisches Werkstück am vorderen Schieber des Förderbandes erkannt
Sensor_SorterCapacitiveSensorPusher2	Sc11	Metallisches Werkstück am hinteren Schieber des Förderbandes erkannt
Sensor_SorterLightbarrierCranInterface	Sc10	Werkstück am Beginn des Förderbandes erkannt
Sensor_SorterLightnessSensorCranInterfaceInverse	Sc11	Helles Werkstück am Beginn des Förderbandes erkannt
Sensor_SorterLightnessSensorPusher1	Sc11	Helles Werkstück am vorderen Schieber des Förderbandes erkannt
Sensor_SorterLightnessSensorPusher2	Sc11	Helles Werkstück am hinteren Schieber des Förderbandes erkannt
Sensor_SorterLightnessSensorPusher3	Sc11	Helles Werkstück am Ende des Förderbandes erkannt

Signal	Vorhanden ab Szenario	Erläuterung
Sensor_SorterPusher1MovedIn	Sc10	Vorderer Schieber am Förderband eingefahren
Sensor_SorterPusher1MovedOut	Sc10	Vorderer Schieber am Förderband ausgefahren
Sensor_SorterPusher2MovedIn	Sc10	Hinterer Schieber am Förderband eingefahren
Sensor_SorterPusher2MovedOut	Sc10	Hinterer Schieber am Förderband ausgefahren
Sensor_StampLowered	Sc3	Stempel in unterer Position (stempelt)
Sensor_StampSliderFilled	Sc3	Werkstück befindet sich im Stempel-Schlitten
Sensor_StampSliderSensorMovedIn	Sc3	Schlitten im Stempel eingefahren
Sensor_StampSliderSensorMovedOut	Sc3	Schlitten im Stempel ausgefahren
Sensor_StampUp	Sc3	Stempel in oberer Position
Sensor_StartButtonMagazin	Sc1	Starttaster am Magazin gedrückt
Sensor_WorkpieceReady	Sc1	Werkstück am Magazin zur Abholung bereit
Sensor_StampStartButton	Sc3	Starttaster am Stempel gedrückt

Tabelle E-2: Informationsmodell der PPU in tabellarischer Darstellung

Signal	Zugehörigkeit zu Equipment	Signaltyp
Actuator_AnalogCranPressure	Stamp	Auxiliary
Actuator_CranLower	Crane	A.Move
Actuator_CranTurnClockwise	Crane	A.Move
Actuator_CranTurnCounterclockwise	Crane	A.Move
Actuator_MagazinGreenLamp	Stack	Auxiliary
Actuator_MagazinSlider	Stack	A.Move
Actuator_MagazinVacuumOff	Crane	A.Hold
Actuator_MagazinVacuumOn	Crane	A.Hold
Actuator_MagazinWhiteLamp	Stack	Auxiliary
Actuator_PusherY1	Conveyor	A.Move
Actuator_PusherY2	Conveyor	A.Move
Actuator_SorterConveyorTowardsStacker	Conveyor	A.Move
Actuator_SorterGreenLampConveyor	Conveyor	Auxiliary
Actuator_StampGreenLamp	Stamp	Auxiliary
Actuator_StampPusher	Stamp	A.WPModify
Actuator_StampSliderMovedIn	Stamp	A.Move
Actuator_StampSliderMovedOut	Stamp	A.Move
Actuator_StampWhiteLamp	Stamp	Auxiliary
Sensor_CranDown	Crane	S.YPosition
Sensor_CranOnConveyor	Crane	S.XPosition
Sensor_CranOnMagazin	Crane	S.XPosition
Sensor_CranPositionStamp	Crane	S.XPosition
Sensor_CranSucked	Crane	S.Hold
Sensor_CranUp	Crane	S.YPosition
Sensor_MagazinCapacitiveSensor	Stack	S.WPIdentify
Sensor_MagazinEmergencyStop	Stack	Auxiliary
Sensor_MagazinOpticalSensor	Stack	S.WPIdentify
Sensor_SliderMovedOut	Stack	S.Xposition
Sensor_SliderNotMovedOut	Stack	S.Xposition
Sensor_SorterCapacitiveSensorPusher1	Conveyor	S.WPIdentify
Sensor_SorterCapacitiveSensorPusher2	Conveyor	S.WPIdentify
Sensor_SorterLightbarrierCranInterface	Conveyor	S.WPDetect
Sensor_SorterLightnessSensorCranInterfaceInverse	Conveyor	S.WPIdentify
Sensor_SorterLightnessSensorPusher1	Conveyor	S.WPIdentify
Sensor_SorterLightnessSensorPusher2	Conveyor	S.WPIdentify
Sensor_SorterLightnessSensorPusher3	Conveyor	S.WPIdentify
Sensor_SorterPusher1MovedIn	Conveyor	S.State
Sensor_SorterPusher1MovedOut	Conveyor	S.Move'
Sensor_SorterPusher2MovedIn	Conveyor	S.State'

Signal	Zugehörigkeit zu Equipment	Signaltyp
Sensor_SorterPusher2MovedOut	Conveyor	S.Move'
Sensor_StampLowered	Stamp	S.Modify'
Sensor_StampSliderFilled	Stamp	S.WPDetect'
Sensor_StampSliderSensorMovedIn	Stamp	S.State'
Sensor_StampSliderSensorMovedOut	Stamp	S.State'
Sensor_StampUp	Stamp	S.Position'
Sensor_StartButtonMagazin	Stack	Auxiliary
Sensor_WorkpieceReady	Stack	S.WPDetect'
Sensor_StampStartButton	Stamp	Auxiliary

## Analyseergebnisse

**Hinweis:** Alle Zeitwerte der Ergebnistabellen sind auf Sekunden gerundet worden. Ausnahmen bilden die Zeiten, die sich auf Operationsdauern beziehen (inkl. der Hauptnutzungszeiten), da diese sich teilweise im Sub-Sekundenbereich befinden.

Tabelle E-3: Analyseergebnisse für Sc1

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen	-	Kein Signal vom Typen A.WPModify vorhanden
Gesamtproduktionsdauer, $t_p$	49 s	
Kran – Belegungszeit, $t_{Crane}^B$	49 s	
Magazin – Belegungszeit, $t_{Stack}^B$	25 s	
Produzierte Menge, $PM$	6	
Minimale Produktionsdauer, $t^{min}$	4 s	
Maximale Produktionsdauer, $t^{max}$	9 s	
Mittlere Produktionsdauer, $t^{\mu}$	7 s	
Anzahl Routen, $AR$	1	
Anzahl Maschinen, $ M $	0	
Prozessfolgen	-	Keine Werkstückbearbeitung
Operationen, $\Omega^w$	-	Keine Werkstückbearbeitung
Gewichtung der Operationen, $g_{\omega}$	-	Keine Werkstückbearbeitung
Anzahl Produkttypen, $ W $	1	
Häufigkeit Zuordnung, $p_{mw}$	-	Keine Werkstückbearbeitung

Tabelle E-4: Analyseergebnisse für Sc3

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen	Stamp	
Positionen der Bearbeitung	Sensor_StampSliderFilled	Ermittelt durch Position, die der Ressource im Informationsmodell zugewiesen ist
Gesamtproduktionsdauer, $t_p$	59 s	
Kran – Belegungszeit, $t_{Crane}^B$	41	
Magazin – Belegungszeit, $t_{Stack}^B$	43 s	
Stempel – Belegungszeit, $t_{Stamp}^B$	6 s	
Dauer der Operation, $e^{Stamp}$	690 ms	Mittel der Dauer des 1-Zustandes von A.WPModify-Signal
Dauer der Operation, $e_1^{Stamp}$	6000 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Hauptnutzungszeit Stempel, $t_{Stamp}^{HN}$	1380 ms	
Produzierte Menge, $PM_0$	4	
Produzierte Menge, $PM_1$	2	

Eigenschaft	Wert	Kommentar
Minimale Produktionsdauer, $t_0^{min}$	4 s	
Minimale Produktionsdauer, $t_1^{min}$	14 s	
Maximale Produktionsdauer, $t_0^{max}$	15 s	
Maximale Produktionsdauer, $t_1^{max}$	15 s	
Mittlere Produktionsdauer, $t_0^\mu$	10 s	
Mittlere Produktionsdauer, $t_1^\mu$	15 s	
Anzahl Routen, $AR_0$	1	
Anzahl Routen, $AR_1$	1	
Anzahl Maschinen, $ M $	1	
Prozessfolgen (Typ 0)	-	Keine Werkstückbearbeitung
Prozessfolgen (Typ 1)	Stamp	
Operationen, $\Omega^0$	-	Keine Werkstückbearbeitung
Operationen, $\Omega^1$	'Actuator_StampPusher'	
Gewichtung der Operation, $g_{Stamp}$	2	Anzahl der Durchführung
Anzahl Produkttypen	2	
Häufigkeit Zuordnung, $p_{Stamp,0}$	-	
Häufigkeit Zuordnung, $p_{Stamp,1}$	100%	

Hinweis: Der Identifikationsvektor besteht aus dem Signal *Sensor\_MagazinCapacitiveSensor*. Somit entspricht Werkstücktyp 0 Kunststoff-Werkstücken und Werkstücktyp 1 metallischen Werkstücken.

Tabelle E-5: Analyseergebnisse für Sc5

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen	Stamp	
Positionen der Bearbeitung	Sensor_StampSliderFilled	Ermittelt durch Position, die der Resource im Informationsmodell zugewiesen ist
Gesamtproduktionsdauer, $t_p$	64 s	
Kran – Belegungszeit, $t_{Crane}^B$	56 s	
Magazin – Belegungszeit, $t_{Stack}^B$	51 s	
Stempel – Belegungszeit, $t_{Stamp}^B$	24 s	
Dauer der Operation, $e^{Stamp}$	691 ms	Mittel der Dauer des 1-Zustandes von A.WPModify-Signal
Dauer der Operation, $e_1^{Stamp}$	12046 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Hauptnutzungszeit Stempel, $t_{Stamp}^{HN}$	1381 ms	
Produzierte Menge, $PM_0$	4	
Produzierte Menge, $PM_1$	2	
Minimale Produktionsdauer, $t_0^{min}$	11 s	
Minimale Produktionsdauer, $t_1^{min}$	21 s	
Maximale Produktionsdauer, $t_0^{max}$	13 s	
Maximale Produktionsdauer, $t_1^{max}$	24 s	
Mittlere Produktionsdauer, $t_0^\mu$	12 s	
Mittlere Produktionsdauer, $t_1^\mu$	22 s	
Anzahl Routen, $AR_0$	1	
Anzahl Routen, $AR_1$	1	
Anzahl Maschinen, $ M $	1	
Prozessfolgen (Typ 0)	-	Keine Werkstückbearbeitung
Prozessfolgen (Typ 1)	Stamp	
Operationen, $\Omega^0$	-	Keine Werkstückbearbeitung
Operationen, $\Omega^1$	'Actuator_StampPusher'	
Gewichtung der Operation, $g_{Stamp}$	2	Anzahl der Durchführung
Anzahl Produkttypen	2	
Häufigkeit Zuordnung, $p_{Stamp,0}$	0	
Häufigkeit Zuordnung, $p_{Stamp,1}$	100%	

Hinweis: Der Identifikationsvektor besteht aus dem Signal *Sensor\_MagazinCapacitiveSensor*. Somit entspricht Werkstücktyp 0 Kunststoff-Werkstücken und Werkstücktyp 1 metallischen Werkstücken.

Tabelle E- 6: Analyseergebnisse für Sc10

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen	Stamp	
Positionen der Bearbeitung	Sensor_StampSliderFilled	Ermittelt durch Position, die der Resource im Informationsmodell zugewiesen ist
Gesamtproduktionsdauer, $t_p$	97 s	
Förderband – Belegungszeit, $t_{Conv}^B$	37 s	
Kran – Belegungszeit, $t_{Crane}^B$	82 s	
Magazin – Belegungszeit, $t_{Stack}^B$	89 s	
Stempel – Belegungszeit, $t_{Stamp}^B$	39 s	
Dauer der Operation, $e^{Stamp}$	686 ms	Mittel der Dauer des 1-Zustandes von A.WPModify-Signal
Dauer der Operation, $e_{11}^{Stamp}$	10689 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Dauer der Operation, $e_{01}^{Stamp}$	3005 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Hauptnutzungszeit Stempel, $t_{Stamp}^{HN}$	3429 ms	
Produzierte Menge, $PM_{00}$	3	
Produzierte Menge, $PM_{01}$	2	
Produzierte Menge, $PM_{11}$	3	
Minimale Produktionsdauer, $t_{00}^{min}$	9 s	
Minimale Produktionsdauer, $t_{01}^{min}$	19 s	
Minimale Produktionsdauer, $t_{11}^{min}$	18 s	
Maximale Produktionsdauer, $t_{00}^{max}$	12 s	
Maximale Produktionsdauer, $t_{01}^{max}$	22 s	
Maximale Produktionsdauer, $t_{11}^{max}$	31 s	
Mittlere Produktionsdauer, $t_{00}^{\mu}$	11 s	
Mittlere Produktionsdauer, $t_{01}^{\mu}$	20 s	
Mittlere Produktionsdauer, $t_{11}^{\mu}$	26 s	
Anzahl Routen, $AR_{00}$	2	1 zusätzliche Route endet inmitten einer anderen
Anzahl Routen, $AR_{01}$	1	1 zusätzliche Route endet inmitten einer anderen
Anzahl Routen, $AR_{11}$	2	1 zusätzliche Route endet inmitten einer anderen
Anzahl Maschinen, $ M $	1	
Prozessfolgen (Typ 00)	-	Keine Werkstückbearbeitung
Prozessfolgen (Typ 01)	Stamp	
Prozessfolgen (Typ 11)	Stamp	
Operationen, $\Omega^{00}$	-	Keine Werkstückbearbeitung
Operationen, $\Omega^{01}$	Actuator_StampPusher	
Operationen, $\Omega^{11}$	Actuator_StampPusher	
Gewichtung der Operation, $g_{Stamp}$	5	Anzahl der Durchführung
Anzahl Produkttypen	3	
Häufigkeit Zuordnung, $p_{Stamp,00}$	0%	
Häufigkeit Zuordnung, $p_{Stamp,01}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,11}$	100%	

Hinweis: Der Identifikationsvektor besteht aus den Signalen [*Sensor\_MagazinCapacitiveSensor* *Sensor\_MagazinOpticalSensor*]. Somit entspricht Werkstücktyp 00 schwarzen Kunststoff-Werkstücken, Werkstücktyp 01 weißen Kunststoff-Werkstücken und Werkstücktyp 11 metallischen Werkstücken.

Tabelle E-7: Analyseergebnisse für Sc11

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen	Stamp	
Positionen der Bearbeitung	Sensor_StampSliderFilled	Ermittelt durch Position, die der Ressource im Informationsmodell zugewiesen ist
Gesamtproduktionsdauer, $t_p$	96,512 s	
Förderband – Belegungszeit, $t_{Conv}^B$	37,384 s	
Kran – Belegungszeit, $t_{Crane}^B$	79,096 s	
Magazin – Belegungszeit, $t_{Stack}^B$	88,9560 s	
Stempel – Belegungszeit, $t_{Stamp}^B$	17,033 s	
Dauer der Operation, $e^{Stamp}$	709 ms	Mittel der Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Stamp-Operation, $e_{1111110}^{Stamp}$	3011 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Dauer Stamp-Operation, $e_{0100100}^{Stamp}$	3426 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Hauptnutzungszeit Stempel, $t_{Stamp}^{HN}$	3543 ms	
Produzierte Menge, $PM_{0000001}$	3	
Produzierte Menge, $PM_{1111110}$	3	
Produzierte Menge, $PM_{0100100}$	2	
Minimale Produktionsdauer, $t_{0000001}^{min}$	3,773 s	
Minimale Produktionsdauer, $t_{1111110}^{min}$	15,069 s	
Minimale Produktionsdauer, $t_{0100100}^{min}$	20,979 s	
Maximale Produktionsdauer, $t_{0000001}^{max}$	6,247 s	
Maximale Produktionsdauer, $t_{1111110}^{max}$	23,729 s	
Maximale Produktionsdauer, $t_{0100100}^{max}$	24,213 s	
Mittlere Produktionsdauer, $t_{0000001}^{\mu}$	5,3246 s	
Mittlere Produktionsdauer, $t_{1111110}^{\mu}$	20,667 s	
Mittlere Produktionsdauer, $t_{0100100}^{\mu}$	21,962 s	
Anzahl Routen, $AR_{0000001}$	1	
Anzahl Routen, $AR_{1111110}$	1	
Anzahl Routen, $AR_{0100100}$	1	
Anzahl Maschinen, $ M $	1	
Prozessfolgen (Typ 0000001)	-	Keine Werkstückbearbeitung
Prozessfolgen (Typ 1111110)	Stamp	
Prozessfolgen (Typ 0100100)	Stamp	
Operationen, $\Omega_{0000001}$	-	Keine Werkstückbearbeitung
Operationen, $\Omega_{1111110}$	'Actuator_StampPusher'	
Operationen, $\Omega_{0100100}$	'Actuator_StampPusher'	
Gewichtung der Operation, $g_{Stamp}$	5	Anzahl der Durchführung
Anzahl Produkttypen	3	
Häufigkeit Zuordnung, $p_{Stamp,0000001}$	0%	
Häufigkeit Zuordnung, $p_{Stamp,1111110}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,0100100}$	100%	

Hinweis: Der Identifikationsvektor besteht aus den Signalen [*Sensor\_MagazinCapacitiveSensor Sensor\_MagazinOpticalSensor Sensor\_SorterCapacitiveSensorPusher1 Sensor\_SorterCapacitiveSensorPusher2 Sensor\_SorterLightnessSensorPusher1 Sensor\_SorterLightnessSensorPusher2 Sensor\_SorterLightnessSensorCranInterfaceInverse*]. Somit entspricht Werkstücktyp 0000001 schwarzen Kunststoff-Werkstücken, Werkstücktyp 0100100 weißen Kunststoff-Werkstücken und Werkstücktyp 1111110 metallischen Werkstücken.

Tabelle E-8: Analyseergebnisse für Sc12

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen	Stamp	
Positionen der Bearbeitung	Sensor_StampSliderFilled	Ermittelt durch Position, die der Ressource im Informationsmodell zugewiesen ist
Gesamtproduktionsdauer, $t_p$	104 s	
Förderband – Belegungszeit, $t_{Conv}^B$	37 s	
Kran – Belegungszeit, $t_{Crane}^B$	97 s	
Magazin – Belegungszeit, $t_{Stack}^B$	98 s	
Stempel – Belegungszeit, $t_{Stamp}^B$	35 s	
Dauer der Operation, $e^{Stamp}$	684 ms	Mittel der Dauer des 1-Zustandes von A.WPModify-Signal
Dauer der Operation, $e_{11111110}^{Stamp}$	3005 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Dauer der Operation, $e_{11111100}^{Stamp}$	3016 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Dauer der Operation, $e_{11101000}^{Stamp}$	3842 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Dauer der Operation, $e_{01001100}^{Stamp}$	12061 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Dauer der Operation, $e_{01001110}^{Stamp}$	12068 ms	Gemessen an mittlerer Dauer, die sich ein Werkstück in der Maschine befand
Hauptnutzungszeit Stempel, $t_{Stamp}^{HN}$	3422 ms	
Produzierte Menge, $PM_{11111110}$	1	
Produzierte Menge, $PM_{11111100}$	1	
Produzierte Menge, $PM_{11101000}$	1	
Produzierte Menge, $PM_{01001110}$	1	
Produzierte Menge, $PM_{01001100}$	1	
Produzierte Menge, $PM_{00000001}$	3	
Minimale Produktionsdauer, $t_{11111110}^{min}$	14 s	
Minimale Produktionsdauer, $t_{11111100}^{min}$	23 s	
Minimale Produktionsdauer, $t_{11101000}^{min}$	23 s	
Minimale Produktionsdauer, $t_{01001110}^{min}$	31 s	
Minimale Produktionsdauer, $t_{01001100}^{min}$	34 s	
Minimale Produktionsdauer, $t_{00000001}^{min}$	4 s	
Maximale Produktionsdauer, $t_{11111110}^{max}$	14 s	
Maximale Produktionsdauer, $t_{11111100}^{max}$	23 s	
Maximale Produktionsdauer, $t_{11101000}^{max}$	23 s	
Maximale Produktionsdauer, $t_{01001110}^{max}$	31 s	
Maximale Produktionsdauer, $t_{01001100}^{max}$	34 s	
Maximale Produktionsdauer, $t_{00000001}^{max}$	14 s	
Mittlere Produktionsdauer, $t_{11111110}^{\mu}$	14 s	
Mittlere Produktionsdauer, $t_{11111100}^{\mu}$	23 s	
Mittlere Produktionsdauer, $t_{11101000}^{\mu}$	23 s	
Mittlere Produktionsdauer, $t_{01001110}^{\mu}$	31 s	
Mittlere Produktionsdauer, $t_{01001100}^{\mu}$	34 s	
Mittlere Produktionsdauer, $t_{00000001}^{\mu}$	10 s	
Anzahl Routen, $AR_{11111110}$	1	
Anzahl Routen, $AR_{11111100}$	1	
Anzahl Routen, $AR_{11101000}$	1	
Anzahl Routen, $AR_{01001110}$	1	
Anzahl Routen, $AR_{01001100}$	1	
Anzahl Routen, $AR_{00000001}$	2	1 zusätzliche Route endet inmitten einer anderen
Anzahl Maschinen, $ M $	1	
Prozessfolgen (Typ 11111110)	Stamp	
Prozessfolgen (Typ 11111100)	Stamp	

Eigenschaft	Wert	Kommentar
Prozessfolgen (Typ 11101000)	Stamp	
Prozessfolgen (Typ 01001110)	Stamp	
Prozessfolgen (Typ 01001100)	Stamp	
Prozessfolgen (Typ 00000001)	-	Keine Werkstückbearbeitung
Operationen, $\Omega^{11111110}$	Actuator_StampPusher	
Operationen, $\Omega^{11111100}$	Actuator_StampPusher	
Operationen, $\Omega^{11101000}$	Actuator_StampPusher	
Operationen, $\Omega^{01001110}$	Actuator_StampPusher	
Operationen, $\Omega^{01001100}$	Actuator_StampPusher	
Operationen, $\Omega^{00000001}$	-	Keine Werkstückbearbeitung
Gewichtung der Operation, $g_{Stamp}$	5	Anzahl der Durchführung
Anzahl Produkttypen	6	Eigentlich 3
Häufigkeit Zuordnung, $p_{Stamp,11111110}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,11111100}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,11101000}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,01001110}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,01001100}$	100%	
Häufigkeit Zuordnung, $p_{Stamp,00000001}$	0%	

Hinweis: Der Identifikationsvektor besteht aus den Signalen [*Sensor\_MagazinCapacitiveSensor Sensor\_MagazinOpticalSensor Sensor\_SorterCapacitiveSensorPusher1 Sensor\_SorterCapacitiveSensorPusher2 Sensor\_SorterLightnessSensorPusher1 Sensor\_SorterLightnessSensorPusher2 Sensor\_SorterLightnessSensorPusher3 Sensor\_SorterLightnessSensorCranInterfaceInverse*]. Somit entsprechen Werkstücktyp 00000001 schwarzen Kunststoff-Werkstücken, die Werkstücktypen 01001110 sowie 01001100 weißen Kunststoff-Werkstücken und die Werkstücktypen 11111110, 11111100 sowie 11101000 metallischen Werkstücken.

---

## **Anhang F: Daten und Ergebnisse zur Anwendung auf der Job-Shop Fertigung**

Auf den folgenden Seiten befinden sich die Prinzipskizzen zu den Teilanlagen der Modellanlage. Hierbei sind alle Signalnamen an den entsprechenden Sensoren und Aktoren angegeben.

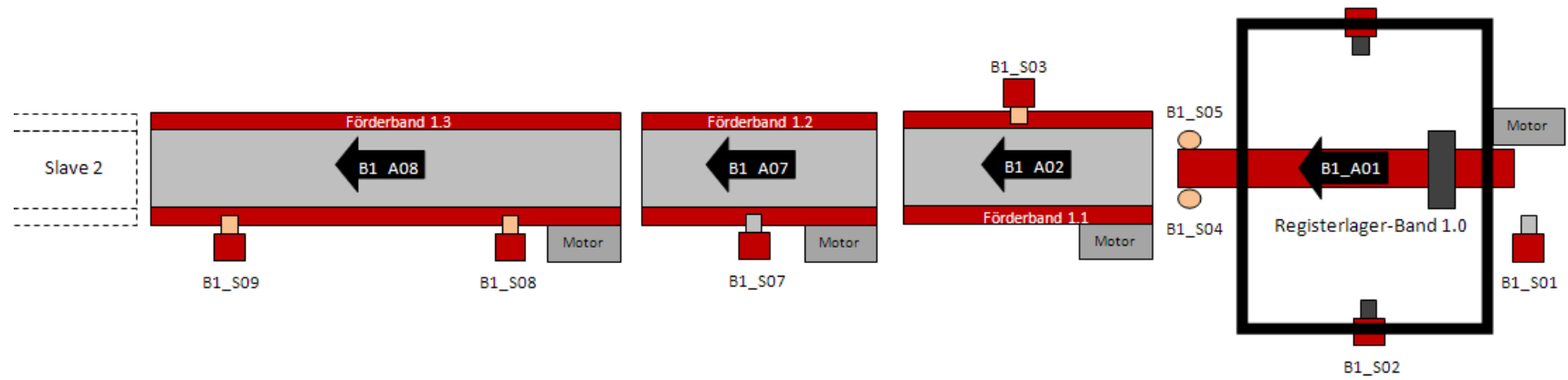


Abbildung F-1: Prinzipskizze der Teilanlage „Slave 1“

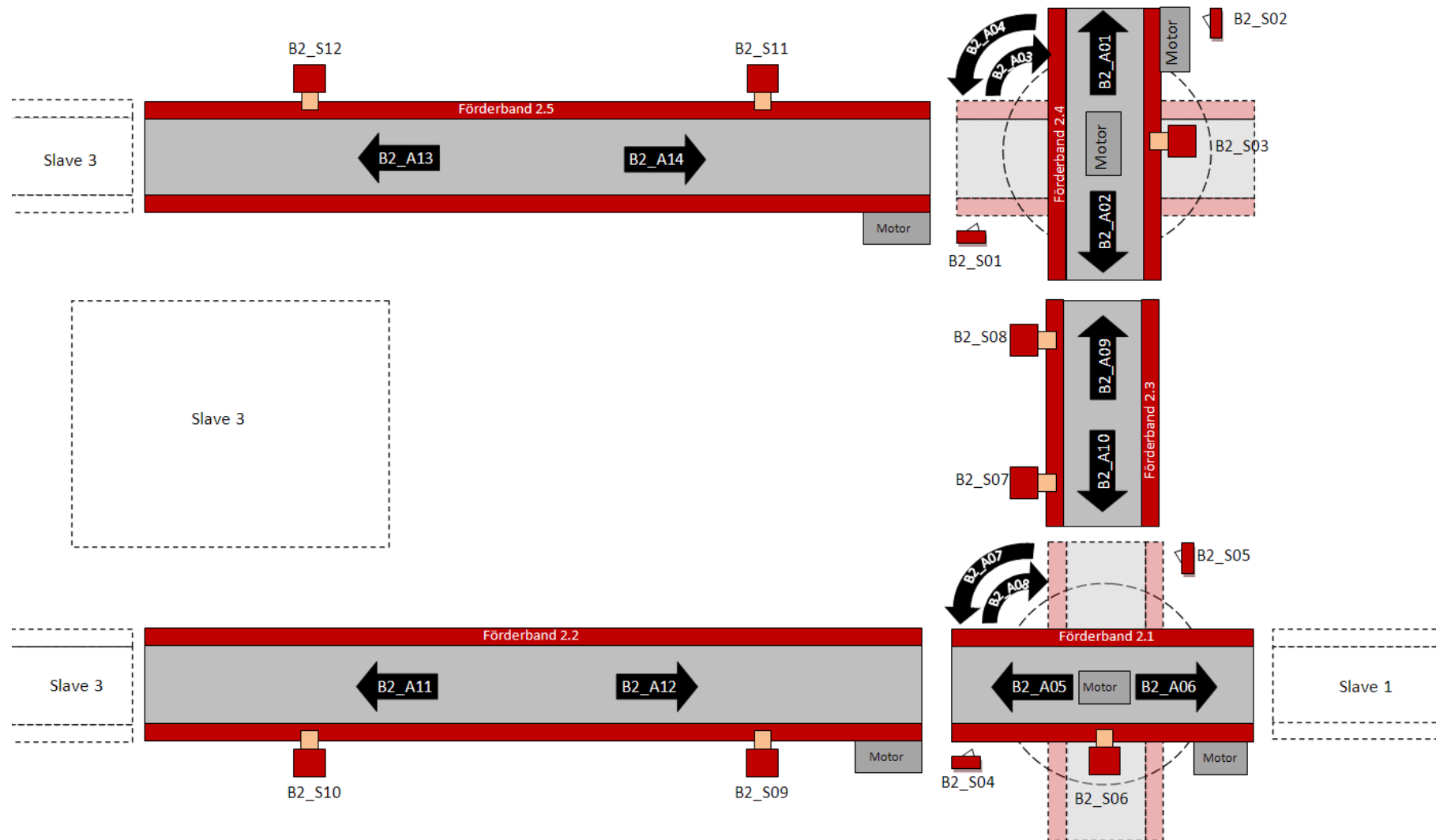


Abbildung F-2: Prinzipskizze der Teilanlage "Slave 2"

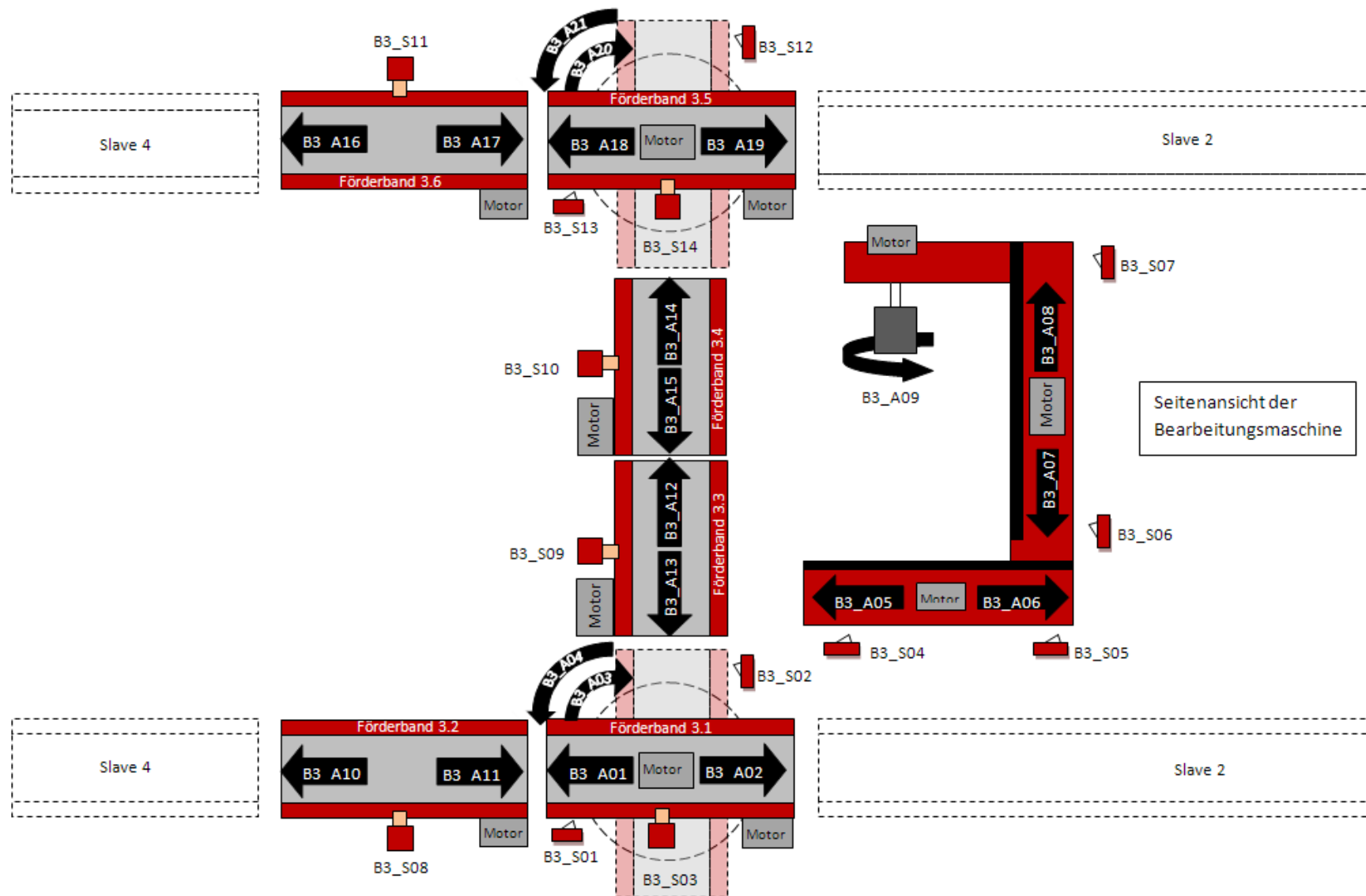


Abbildung F-3: Prinzipskizze der Teilanlage „Slave 3“

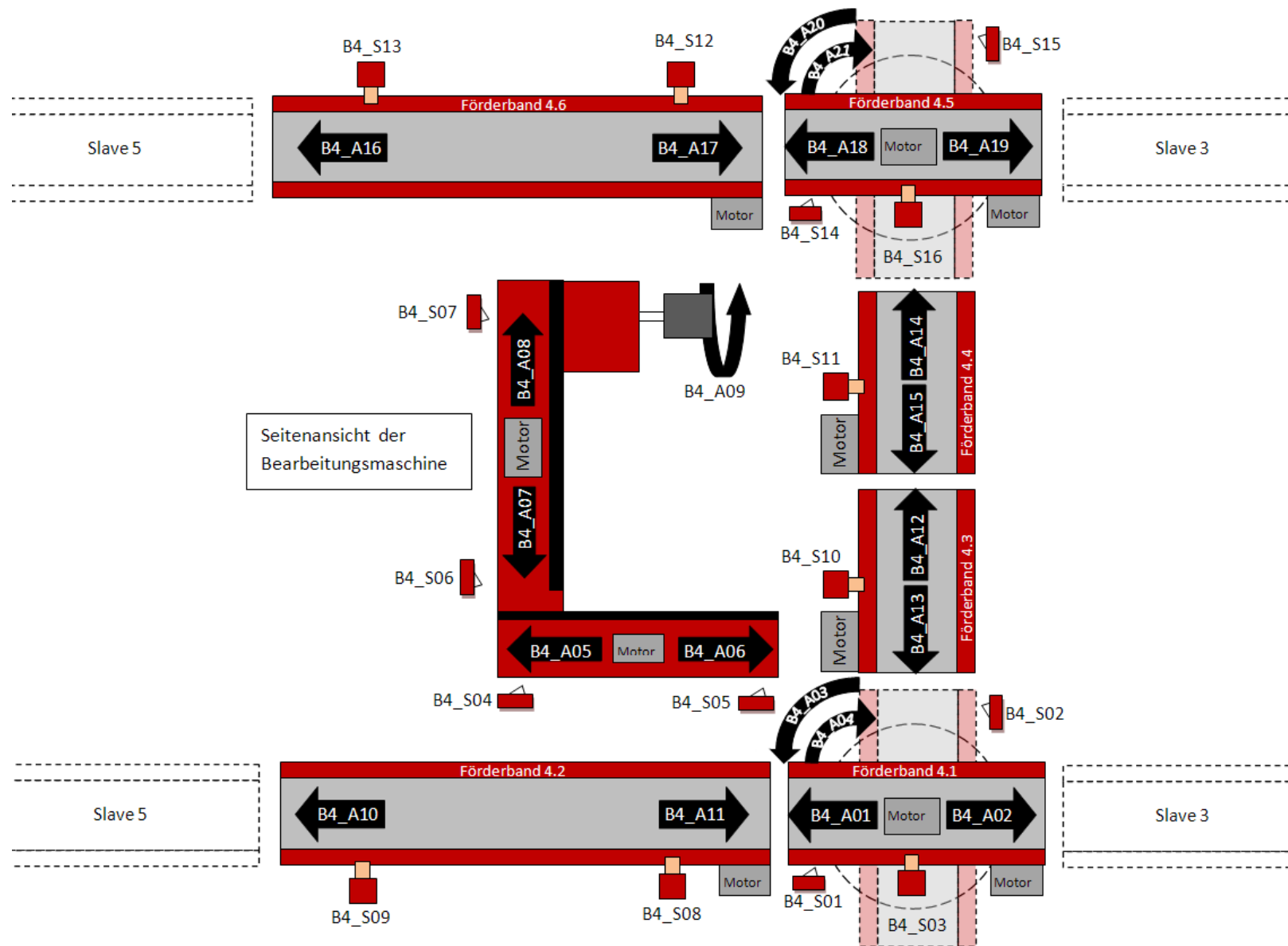


Abbildung F-4: Prinzipskizze der Teilanlage "Slave 4"



Tabelle F-1: I/O-Signale der Anlage

Signal	Beschreibung
B1_A01	Registerlager-Band 1.0
B1_A02	Förder-Band 1.1
B1_A03	LED
B1_A04	Summer
B1_A05	Farbcode Bit 0
B1_A06	Farbcode Bit1
B1_A07	Förder-Band 1.2
B1_A08	Förder-Band 1.3
B1_S01	Reedkontakt Registerlager
B1_S02	Lichtschanke Registerlager
B1_S03	Initiator Förder-Band 1.1
B1_S04	Reedkontakt Farberkennung rechts
B1_S05	Reedkontakt Farberkennung rechts
B1_S06	Handtaster Registerlager
B1_S07	Initiator Förder-Band 1.2
B1_S08	Initiator Förder-Band 1.3 rechts
B1_S09	Initiator Förder-Band 1.3 links
B2_A01	Drehtisch Förderband 2.4
B2_A02	Drehtisch Förderband 2.4
B2_A03	Drehtisch Drehmotor (rechtsrum)
B2_A04	Drehtisch Drehmotor (linksrum)
B2_A05	Drehtisch Förderband 2.1
B2_A06	Drehtisch Förderband 2.1
B2_A07	Drehtisch Drehmotor (linksrum)
B2_A08	Drehtisch Drehmotor (rechtsrum)
B2_A09	Förderband 2.3 Richtung Förderband 2.4
B2_A10	Förderband 2.3 Richtung Förderband 2.1
B2_A11	Förderband 2.2 Richtung Slave 3
B2_A12	Förderband 2.2 Richtung Förderband 2.1
B2_A13	Förderband 2.5 Richtung Slave 3
B2_A14	Förderband 2.5 Richtung Förderband 2.4
B2_S01	Drehtisch in Position (Übergabe Förderband 2.5)
B2_S02	Drehtisch in Position (Übergabe Förderband 2.3)
B2_S03	Drehtisch Förderband 2.4 - Initiator
B2_S04	Drehtisch in Position (Übernahme Slave1)
B2_S05	Drehtisch in Position (Übergabe Förderband 2.3)
B2_S06	Drehtisch Förderband 2.1 - Initiator
B2_S07	Förderband 2.3 – Initiator
B2_S08	Förderband 2.3 – Initiator
B2_S09	Förderband 2.2 – Initiator
B2_S10	Förderband 2.2 – Initiator
B2_S11	Förderband 2.5 – Initiator
B2_S12	Förderband 2.5 – Initiator
B3_A01	Drehtisch Förderband 3.1
B3_A02	Drehtisch Förderband 3.1
B3_A03	Drehtisch Drehmotor (rechtsrum)
B3_A04	Drehtisch Drehmotor (linksrum)
B3_A05	Bearbeitungsmaschine nach vorn
B3_A06	Bearbeitungsmaschine nach hinten
B3_A07	Bearbeitungsmaschine nach unten
B3_A08	Bearbeitungsmaschine nach oben
B3_A09	Werkzeug drehen
B3_A10	Förderband 3.2 Richtung Slave 4
B3_A11	Förderband 3.2 Richtung Förderband 3.1
B3_A12	Förderband 3.3 Richtung Förderband 3.4
B3_A13	Förderband 3.3 Richtung Förderband 3.1
B3_A14	Förderband 3.4 Richtung Förderband 3.5
B3_A15	Förderband 3.4 Richtung Förderband 3.3
B3_A16	Förderband 3.6 Richtung Slave 4
B3_A17	Förderband 3.6 Richtung Förderband 3.5
B3_A18	Drehtisch Förderband 3.5
B3_A19	Drehtisch Förderband 3.5
B3_A20	Drehtisch Drehmotor (rechtsrum)

<b>Signal</b>	<b>Beschreibung</b>
B3_A21	Drehtisch Drehmotor (linksrum)
B3_S01	Drehtisch in Position (Übernahme Slave 2)
B3_S02	Drehtisch in Position (Übergabe Förderband 3.3)
B3_S03	Drehtisch Förderband 3.1 - Initiator
B3_S04	Endschalter Bearbeitungsm., vorn
B3_S05	Endschalter Bearbeitungsm., hinten
B3_S06	Endschalter Bearbeitungsm., unten
B3_S07	Endschalter Bearbeitungsm., oben
B3_S08	Förderband 3.2 – Initiator
B3_S09	Förderband 3.3 – Initiator
B3_S10	Förderband 3.4 – Initiator
B3_S11	Förderband 3.6 – Initiator
B3_S12	Drehtisch in Position (Übergabe Förderband 3.4)
B3_S13	Drehtisch in Position (Übernahme Slave 2)
B3_S14	Drehtisch Förderband 4.5 – Initiator
B4_A01	Drehtisch Förderband 5.1
B4_A02	Drehtisch Förderband 5.1
B4_A03	Drehtisch Drehmotor (linksrum)
B4_A04	Drehtisch Drehmotor (rechtsrum)
B4_A05	Bearbeitungsmaschine nach hinten
B4_A06	Bearbeitungsmaschine nach vorn
B4_A07	Bearbeitungsmaschine nach unten
B4_A08	Bearbeitungsmaschine nach oben
B4_A09	Werkzeug drehen
B4_A10	Förderband 4.2 Richtung Slave 5
B4_A11	Förderband 4.2 Richtung Förderband 4.1
B4_A12	Förderband 4.3 Richtung Förderband 4.4
B4_A13	Förderband 4.3 Richtung Förderband 4.1
B4_A14	Förderband 4.4 Richtung Förderband 4.5
B4_A15	Förderband 4.4 Richtung Förderband 4.3
B4_A16	Förderband 4.6 Richtung Slave 5
B4_A17	Förderband 4.6 Richtung Förderband 4.5
B4_A18	Drehtisch Förderband 4.5
B4_A19	Drehtisch Förderband 4.5
B4_A20	Drehtisch Drehmotor (linksrum)
B4_A21	Drehtisch Drehmotor (rechtsrum)
B4_S01	Drehtisch in Position (Übernahme Slave 3)
B4_S02	Drehtisch in Position (Übergabe Förderband 4.3)
B4_S03	Drehtisch Förderband 4.1 - Initiator
B4_S04	Endschalter Bearbeitungsm., hinten
B4_S05	Endschalter Bearbeitungsm.,vorn
B4_S06	Endschalter Bearbeitungsm., unten
B4_S07	Endschalter Bearbeitungsm., oben
B4_S08	Förderband 4.2 – Initiator
B4_S09	Förderband 4.2 – Initiator
B4_S10	Förderband 4.3 – Initiator
B4_S11	Förderband 4.4 – Initiator
B4_S12	Förderband 4.6 – Initiator
B4_S13	Förderband 4.6 – Initiator
B4_S14	Drehtisch in Position (Übernahme Slave 3)
B4_S15	Drehtisch in Position (Übergabe Förderband 4.4)
B4_S16	Drehtisch Förderband 4.5 – Initiator
B5_A01	Drehtisch Förderband 5.1
B5_A02	Drehtisch Förderband 5.1
B5_A03	Drehtisch Drehmotor (rechtsrum)
B5_A04	Drehtisch Drehmotor (linksrum)
B5_A05	Bearbeitungsmaschine nach vorn
B5_A06	Bearbeitungsmaschine nach hinten
B5_A07	Bearbeitungsmaschine nach unten
B5_A08	Bearbeitungsmaschine nach oben
B5_A09	Werkzeug drehen
B5_A10	Werkzeugkopf drehen
B5_A11	Förderband 5.3 Richtung Förderband 5.4
B5_A12	Förderband 5.3 Richtung Förderband 5.2
B5_A13	Förderband 5.2 Richtung Förderband 5.3

Signal	Beschreibung
B5_A14	Förderband 5.2 Richtung Förderband 5.1
B5_A15	Drehtisch Förderband 5.4
B5_A16	Drehtisch Förderband 5.4
B5_A17	Drehtisch Drehmotor (rechtsrum)
B5_A18	Drehtisch Drehmotor (linksrum)
B5_S01	Drehtisch in Position (Übernahme Slave 4)
B5_S02	Drehtisch in Position (Übergabe Förderband 5.2)
B5_S03	Drehtisch Förderband 5.1 - Initiator
B5_S04	Endschalter Bearbeitungsm., hinten
B5_S05	Endschalter Bearbeitungsm., vorn
B5_S06	Endschalter Bearbeitungsm., unten
B5_S07	Endschalter Bearbeitungsm., oben
B5_S08	Endschalter Werkzeugkopf in Arbeitsstellung
B5_S09	Förderband 5.3 – Initiator
B5_S10	Förderband 5.2 – Initiator
B5_S11	Drehtisch in Position (Übernahme Slave4)
B5_S12	Drehtisch in Position (Übergabe Förderband 5.3)
B5_S13	Drehtisch Förderband 5.4 – Initiator

Tabelle F-2: Informationsmodell der Job-Shop-Fertigung

Signal	Zugehörigkeit zu Equipment	Signaltyp
B1_A01	Slave1_Conv1_0	A.Move
B1_A02	Slave1_Conv1_1	A.Move
B1_A03	Slave1	Auxiliary
B1_A04	Slave1	Auxiliary
B1_A05	Slave1	Auxiliary
B1_A06	Slave1	Auxiliary
B1_A07	Slave1_Conv1_2	A.Move
B1_A08	Slave1_Conv1_3	A.Move
B1_S01	Slave1_Conv1_0	S.State
B1_S02	Slave1_Conv1_0	S.WPDetect
B1_S03	Slave1_Conv1_1	S.WPDetect
B1_S04	Slave1_Conv1_0	S.WPIdentify
B1_S05	Slave1_Conv1_0	S.WPIdentify
B1_S06	Slave1	Auxiliary
B1_S07	Slave1_Conv1_2	S.WPDetect
B1_S08	Slave1_Conv1_3	S.WPDetect
B1_S09	Slave1_Conv1_3	S.WPDetect
B2_A01	Slave2_Table2_4	A.Move
B2_A02	Slave2_Table2_4	A.Move
B2_A03	Slave2_Table2_4	A.Move
B2_A04	Slave2_Table2_4	A.Move
B2_A05	Slave2_Table2_1	A.Move
B2_A06	Slave2_Table2_1	A.Move
B2_A07	Slave2_Table2_1	A.Move
B2_A08	Slave2_Table2_1	A.Move
B2_A09	Slave2_Conv2_3	A.Move
B2_A10	Slave2_Conv2_3	A.Move
B2_A11	Slave2_Conv2_2	A.Move
B2_A12	Slave2_Conv2_2	A.Move
B2_A13	Slave2_Conv2_5	A.Move
B2_A14	Slave2_Conv2_5	A.Move
B2_S01	Slave2_Table2_4	S.State
B2_S02	Slave2_Table2_4	S.State
B2_S03	Slave2_Table2_4	S.WPDetect
B2_S04	Slave2_Table2_1	S.State
B2_S05	Slave2_Table2_1	S.State
B2_S06	Slave2_Table2_1	S.WPDetect
B2_S07	Slave2_Conv2_3	S.WPDetect

Signal	Zugehörigkeit zu Equipment	Signaltyp
B2_S08	Slave2_Conv2_3	S.WPDetect
B2_S09	Slave2_Conv2_2	S.WPDetect
B2_S10	Slave2_Conv2_2	S.WPDetect
B2_S11	Slave2_Conv2_5	S.WPDetect
B2_S12	Slave2_Conv2_5	S.WPDetect
B3_A01	Slave3_Table3_1	A.Move
B3_A02	Slave3_Table3_1	A.Move
B3_A03	Slave3_Table3_1	A.Move
B3_A04	Slave3_Table3_1	A.Move
B3_A05	Slave3_M3_1	A.Move
B3_A06	Slave3_M3_1	A.Move
B3_A07	Slave3_M3_1	A.Move
B3_A08	Slave3_M3_1	A.Move
B3_A09	Slave3_M3_1	A.WPModify
B3_A10	Slave3_Conv3_2	A.Move
B3_A11	Slave3_Conv3_2	A.Move
B3_A12	Slave3_Conv3_3	A.Move
B3_A13	Slave3_Conv3_3	A.Move
B3_A14	Slave3_Conv3_4	A.Move
B3_A15	Slave3_Conv3_4	A.Move
B3_A16	Slave3_Conv3_6	A.Move
B3_A17	Slave3_Conv3_6	A.Move
B3_A18	Slave3_Table3_5	A.Move
B3_A19	Slave3_Table3_5	A.Move
B3_A20	Slave3_Table3_5	A.Move
B3_A21	Slave3_Table3_5	A.Move
B3_S01	Slave3_Table3_1	S.State
B3_S02	Slave3_Table3_1	S.State
B3_S03	Slave3_Table3_1	S.WPDetect
B3_S04	Slave3_M3_1	S.State
B3_S05	Slave3_M3_1	S.State
B3_S06	Slave3_M3_1	S.State
B3_S07	Slave3_M3_1	S.State
B3_S08	Slave3_Conv3_2	S.WPDetect
B3_S09	Slave3_Conv3_3	S.WPDetect
B3_S10	Slave3_Conv3_4	S.WPDetect
B3_S11	Slave3_Conv3_6	S.WPDetect
B3_S12	Slave3_Table3_5	S.State
B3_S13	Slave3_Table3_5	S.State
B3_S14	Slave3_Table3_5	S.WPDetect
B4_A01	Slave4_Table4_1	A.Move
B4_A02	Slave4_Table4_1	A.Move
B4_A03	Slave4_Table4_1	A.Move
B4_A04	Slave4_Table4_1	A.Move
B4_A05	Slave4_M4_2	A.Move
B4_A06	Slave4_M4_2	A.Move
B4_A07	Slave4_M4_2	A.Move
B4_A08	Slave4_M4_2	A.Move
B4_A09	Slave4_M4_2	A.WPModify
B4_A10	Slave4_Conv4_2	A.Move
B4_A11	Slave4_Conv4_2	A.Move
B4_A12	Slave4_Conv4_3	A.Move
B4_A13	Slave4_Conv4_3	A.Move
B4_A14	Slave4_Conv4_4	A.Move
B4_A15	Slave4_Conv4_4	A.Move
B4_A16	Slave4_Conv4_6	A.Move
B4_A17	Slave4_Conv4_6	A.Move
B4_A18	Slave4_Table4_5	A.Move
B4_A19	Slave4_Table4_5	A.Move
B4_A20	Slave4_Table4_5	A.Move
B4_A21	Slave4_Table4_5	A.Move

Signal	Zugehörigkeit zu Equipment	Signaltyp
B4_S01	Slave4_Table4_1	S.State
B4_S02	Slave4_Table4_1	S.State
B4_S03	Slave4_Table4_1	S.WPDetect
B4_S04	Slave4_M4_2	S.State
B4_S05	Slave4_M4_2	S.State
B4_S06	Slave4_M4_2	S.State
B4_S07	Slave4_M4_2	S.State
B4_S08	Slave4_Conv4_2	S.WPDetect
B4_S09	Slave4_Conv4_2	S.WPDetect
B4_S10	Slave4_Conv4_3	S.WPDetect
B4_S11	Slave4_Conv4_4	S.WPDetect
B4_S12	Slave4_Conv4_6	S.WPDetect
B4_S13	Slave4_Conv4_6	S.WPDetect
B4_S14	Slave4_Table4_5	S.State
B4_S15	Slave4_Table4_5	S.State
B4_S16	Slave4_Table4_5	S.WPDetect
B5_A01	Slave5_Table5_1	A.Move
B5_A02	Slave5_Table5_1	A.Move
B5_A03	Slave5_Table5_1	A.Move
B5_A04	Slave5_Table5_1	A.Move
B5_A05	Slave5_M5_3	A.Move
B5_A06	Slave5_M5_3	A.Move
B5_A07	Slave5_M5_3	A.Move
B5_A08	Slave5_M5_3	A.Move
B5_A09	Slave5_M5_3	A.Toolchange
B5_A10	Slave5_M5_3	A.WPModify
B5_A11	Slave5_Table5_3	A.Move
B5_A12	Slave5_Table5_3	A.Move
B5_A13	Slave5_Table5_2	A.Move
B5_A14	Slave5_Table5_2	A.Move
B5_A15	Slave5_Table5_4	A.Move
B5_A16	Slave5_Table5_4	A.Move
B5_A17	Slave5_Table5_4	A.Move
B5_A18	Slave5_Table5_4	A.Move
B5_S01	Slave5_Table5_1	S.State
B5_S02	Slave5_Table5_1	S.State
B5_S03	Slave5_Table5_1	S.WPDetect
B5_S04	Slave5_M5_3	S.State
B5_S05	Slave5_M5_3	S.State
B5_S06	Slave5_M5_3	S.State
B5_S07	Slave5_M5_3	S.State
B5_S08	Slave5_M5_3	S.State
B5_S09	Slave5_Table5_3	S.WPDetect
B5_S10	Slave5_Table5_2	S.WPDetect
B5_S11	Slave5_Table5_4	S.State
B5_S12	Slave5_Table5_4	S.State
B5_S13	Slave5_Table5_4	S.WPDetect

Tabelle F-3: Analyseergebnisse für Sc1

Eigenschaft	Wert	Kommentar	
Werkstück bearbeitende Ressourcen u. Signale u. Positionen	Slave3_M3_1 B3_A09 B3_S09'	Maschine M1	
	Slave4_M4_2 B4_A09 B4_S10	Maschine M2	
	Slave5_M5_3 B5_A10' B5_S10	Maschine M3	
	Gesamtproduktionsdauer, $t_P$	631 s	
	M1 – Belegungszeit, $t_{M1}^B$	46 s	
	M2 - Belegungszeit, $t_{M2}^B$	42 s	
	M3 - Belegungszeit, $t_{M3}^B$	42 s	
	Dauer Operation an M1, $e^{M1}$	8 s	Gemessen an Dauer des High-Zustandes von A.WPModify-Signal
	Dauer Operation an M1, $e_{01}^{M1}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{11}^{M1}$	9 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M1, $e_{12}^{M1}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M1, $e_{22}^{M1}$	9 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Hauptnutzungszeit M1, $t_{M1}^{HN}$	32 s		
Dauer Operation an M2, $e^{M2}$	7 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal	
Dauer Operation an M2, $e_{01}^{M2}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M2, $e_{11}^{M2}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M2, $e_{12}^{M2}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M2, $e_{22}^{M2}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Hauptnutzungszeit M2, $t_{M2}^{HN}$	27 s		
Dauer Operation an M3, $e^{M3}$	6 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal	
Dauer Operation an M3, $e_{01}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M3, $e_{11}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M3, $e_{12}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Dauer Operation an M3, $e_{22}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand	
Hauptnutzungszeit M3, $t_{M3}^{HN}$	22 s		
Produzierte Menge, $PM_{00}$	3		
Produzierte Menge, $PM_{01}$	1		
Produzierte Menge, $PM_{11}$	1		
Produzierte Menge, $PM_{12}$	1		
Produzierte Menge, $PM_{22}$	1		
Minimale Produktionsdauer, $t_{00}^{min}$	29 s		
Minimale Produktionsdauer, $t_{01}^{min}$	148 s		
Minimale Produktionsdauer, $t_{11}^{min}$	91 s		
Minimale Produktionsdauer, $t_{12}^{min}$	186 s		

Eigenschaft	Wert	Kommentar
Minimale Produktionsdauer, $t_{22}^{min}$	102 s	
Maximale Produktionsdauer, $t_{00}^{max}$	111 s	
Maximale Produktionsdauer, $t_{01}^{max}$	148 s	
Maximale Produktionsdauer, $t_{11}^{max}$	91 s	
Maximale Produktionsdauer, $t_{12}^{max}$	186 s	
Maximale Produktionsdauer, $t_{22}^{max}$	102 s	
Mittlere Produktionsdauer, $t_{00}^{\mu}$	65 s	
Mittlere Produktionsdauer, $t_{01}^{\mu}$	148 s	
Mittlere Produktionsdauer, $t_{11}^{\mu}$	91 s	
Mittlere Produktionsdauer, $t_{12}^{\mu}$	186 s	
Mittlere Produktionsdauer, $t_{22}^{\mu}$	102 s	
Anzahl Routen, $AR_{00}$	1	
Anzahl Routen, $AR_{01}$	1	
Anzahl Routen, $AR_{11}$	1	
Anzahl Routen, $AR_{12}$	1	
Anzahl Routen, $AR_{22}$	1	
Anzahl Maschinen, $ M $	3	
Prozessfolgen (Typ 00)	-	Keine Bearbeitung
Prozessfolgen (Typ 01)	Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Prozessfolgen (Typ 11)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Prozessfolgen (Typ 12)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Prozessfolgen (Typ 22)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Operationen, $\Omega^{00}$	-	Keine Bearbeitung
Operationen, $\Omega^{01}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{11}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{12}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{22}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Gewichtung der Operation durch M1, $g_{M1}$	4	Anzahl an Durchführungen
Gewichtung der Operation durch M2, $g_{M2}$	4	Anzahl an Durchführungen
Gewichtung der Operation durch M3, $g_{M3}$	4	Anzahl an Durchführungen
Anzahl Produkttypen	5	
Häufigkeit Zuordnung, $p_{M1,00}$	0%	
Häufigkeit Zuordnung, $p_{M2,00}$	0%	
Häufigkeit Zuordnung, $p_{M3,00}$	0%	
Häufigkeit Zuordnung, $p_{M1,01}$	100%	
Häufigkeit Zuordnung, $p_{M2,01}$	100%	
Häufigkeit Zuordnung, $p_{M3,01}$	100%	
Häufigkeit Zuordnung, $p_{M1,11}$	100%	
Häufigkeit Zuordnung, $p_{M2,11}$	100%	

Eigenschaft	Wert	Kommentar
Häufigkeit Zuordnung, $p_{M3,11}$	100%	
Häufigkeit Zuordnung, $p_{M1,12}$	100%	
Häufigkeit Zuordnung, $p_{M2,12}$	100%	
Häufigkeit Zuordnung, $p_{M3,12}$	100%	
Häufigkeit Zuordnung, $p_{M1,22}$	100%	
Häufigkeit Zuordnung, $p_{M2,22}$	100%	
Häufigkeit Zuordnung, $p_{M3,22}$	100%	

Hinweis: Der Identifikationsvektor setzt sich aus der Häufigkeit der Auslösung der beiden Reed-Kontakte in Slave 1 durch ein Werkstück des jeweiligen Typs zusammen. Somit entspricht der Vektor [00] naturfarbenen Werkstücken, [01] gelben Werkstücken, [11] roten Werkstücken, [12] grünen Werkstücken und [22] blauen Werkstücken. Dies gilt auch für die folgenden Ergebnistabellen.

TabelleF-4: Analyseergebnisse für Sc2

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen u. Signale u. Positionen	Slave3_M3_1 B3_A09 B3_S09	Maschine M1
	Slave4_M4_2 B4_A09 B4_S10	Maschine M2
	Slave5_M5_3 B5_A10 B5_S10	Maschine M3
Gesamtproduktionsdauer, $t_p$	583 s	
M1 – Belegungszeit, $t_{M1}^B$	52 s	
M2 - Belegungszeit, $t_{M2}^B$	86 s	
M3 - Belegungszeit, $t_{M3}^B$	50 s	
Dauer Operation an M1, $e_{00}^{M1}$	8 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Operation an M1, $e_{01}^{M1}$	20 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{11}^{M1}$	13 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{12}^{M1}$	14 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{22}^{M1}$	20 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Hauptnutzungszeit M1, $t_{M1}^{HN}$	34 s	
Dauer Operation an M2, $e_{00}^{M2}$	4 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Operation an M2, $e_{01}^{M2}$	9 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M2, $e_{11}^{M2}$	9 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M2, $e_{12}^{M2}$	9 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M2, $e_{22}^{M2}$	9 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Hauptnutzungszeit M2, $t_{M2}^{HN}$	16 s	
Dauer Operation an M3, $e_{00}^{M3}$	3 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Operation an M3, $e_{01}^{M3}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M3, $e_{11}^{M3}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M3, $e_{12}^{M3}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand

Eigenschaft	Wert	Kommentar
Dauer Operation an M3, $e_{12}^{M3}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M3, $e_{22}^{M3}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Hauptnutzungszeit M3, $t_{M3}^{HN}$	18 s	
Produzierte Menge, $PM_{00}$	3	
Produzierte Menge, $PM_{01}$	1	
Produzierte Menge, $PM_{11}$	1	
Produzierte Menge, $PM_{12}$	1	
Produzierte Menge, $PM_{22}$	1	
Minimale Produktionsdauer, $t_{00}^{min}$	61 s	
Minimale Produktionsdauer, $t_{01}^{min}$	249 s	
Minimale Produktionsdauer, $t_{11}^{min}$	195 s	
Minimale Produktionsdauer, $t_{12}^{min}$	226 s	
Minimale Produktionsdauer, $t_{22}^{min}$	187 s	
Maximale Produktionsdauer, $t_{00}^{max}$	103 s	
Maximale Produktionsdauer, $t_{01}^{max}$	249 s	
Maximale Produktionsdauer, $t_{11}^{max}$	195 s	
Maximale Produktionsdauer, $t_{12}^{max}$	226 s	
Maximale Produktionsdauer, $t_{22}^{max}$	187 s	
Mittlere Produktionsdauer, $t_{00}^{\mu}$	80 s	
Mittlere Produktionsdauer, $t_{01}^{\mu}$	249 s	
Mittlere Produktionsdauer, $t_{11}^{\mu}$	195 s	
Mittlere Produktionsdauer, $t_{12}^{\mu}$	226 s	
Mittlere Produktionsdauer, $t_{22}^{\mu}$	187 s	
Anzahl Routen, $AR_{00}$	1	
Anzahl Routen, $AR_{01}$	1	
Anzahl Routen, $AR_{11}$	1	
Anzahl Routen, $AR_{12}$	1	
Anzahl Routen, $AR_{22}$	1	
Anzahl Maschinen, $ M $	3	
Prozessfolgen (Typ 00)	'Slave5_M5_3'	Erste Bearbeitung
Prozessfolgen (Typ 01)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Prozessfolgen (Typ 11)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Prozessfolgen (Typ 12)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Prozessfolgen (Typ 22)	'Slave3_M3_1'	Erste Bearbeitung
	'Slave4_M4_2'	Zweite Bearbeitung
	'Slave5_M5_3'	Dritte Bearbeitung
Operationen, $\Omega^{00}$	{Slave5_M5_3}	
Operationen, $\Omega^{01}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{11}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{12}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	

Eigenschaft	Wert	Kommentar
Operationen, $\Omega^{22}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Gewichtung der Operation durch M1, $g_{M1}$	4	Anzahl an Durchführungen
Gewichtung der Operation durch M2, $g_{M2}$	4	Anzahl an Durchführungen
Gewichtung der Operation durch M3, $g_{M3}$	7	Anzahl an Durchführungen
Anzahl Produkttypen	5	
Häufigkeit Zuordnung, $p_{M1,00}$	0%	
Häufigkeit Zuordnung, $p_{M2,00}$	0%	
Häufigkeit Zuordnung, $p_{M3,00}$	100%	
Häufigkeit Zuordnung, $p_{M1,01}$	100%	
Häufigkeit Zuordnung, $p_{M2,01}$	100%	
Häufigkeit Zuordnung, $p_{M3,01}$	100%	
Häufigkeit Zuordnung, $p_{M1,11}$	100%	
Häufigkeit Zuordnung, $p_{M2,11}$	100%	
Häufigkeit Zuordnung, $p_{M3,11}$	100%	
Häufigkeit Zuordnung, $p_{M1,12}$	100%	
Häufigkeit Zuordnung, $p_{M2,12}$	100%	
Häufigkeit Zuordnung, $p_{M3,12}$	100%	
Häufigkeit Zuordnung, $p_{M1,22}$	100%	
Häufigkeit Zuordnung, $p_{M2,22}$	100%	
Häufigkeit Zuordnung, $p_{M3,22}$	100%	

Tabelle F-5: Analyseergebnisse für Sc3

Eigenschaft	Wert	Kommentar
Werkstück bearbeitende Ressourcen u. Signale u. Positionen	Slave3_M3_1 B3_A09 B3_S09	Maschine M1
	Slave4_M4_2 B4_A09 B4_S10	Maschine M2
	Slave5_M5_3 B5_A10 B5_S10	Maschine M3
Gesamtproduktionsdauer, $t_p$	819 s	
M1 – Belegungszeit, $t_{M1}^B$	89 s	
M2 - Belegungszeit, $t_{M2}^B$	46 s	
M3 - Belegungszeit, $t_{M3}^B$	38 s	
Dauer Operation an M1, $e^{M1}$	8 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Operation an M1, $e_{01}^{M1}$	16 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{11}^{M1}$	12 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{12}^{M1}$	20 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M1, $e_{22}^{M1}$	12 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Hauptnutzungszeit M1, $t_{M1}^{HN}$	49 s	
Dauer Operation an M2, $e^{M2}$	4 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Operation an M2, $e_{01}^{M2}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M2, $e_{11}^{M2}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand

Eigenschaft	Wert	Kommentar
		ne befand
Dauer Operation an M2, $e_{12}^{M2}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M2, $e_{22}^{M2}$	8 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Hauptnutzungszeit M2, $t_{M2}^{HN}$	23 s	
Dauer Operation an M3, $e^{M3}$	3 s	Gemessen an Dauer des 1-Zustandes von A.WPModify-Signal
Dauer Operation an M3, $e_{00}^{M3}$	0,78 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand - FALSCH
Dauer Operation an M3, $e_{01}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M3, $e_{11}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M3, $e_{12}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Dauer Operation an M3, $e_{22}^{M3}$	7 s	Gemessen an Dauer, die sich ein Werkstück an der Maschine befand
Hauptnutzungszeit M3, $t_{M3}^{HN}$	15 s	
Produzierte Menge, $PM_{00}$	4	
Produzierte Menge, $PM_{01}$	3	
Produzierte Menge, $PM_{11}$	1	
Produzierte Menge, $PM_{12}$	1	
Produzierte Menge, $PM_{22}$	1	
Minimale Produktionsdauer, $t_{00}^{min}$	54 s	
Minimale Produktionsdauer, $t_{01}^{min}$	204 s	
Minimale Produktionsdauer, $t_{11}^{min}$	197 s	
Minimale Produktionsdauer, $t_{12}^{min}$	219 s	
Minimale Produktionsdauer, $t_{22}^{min}$	230 s	
Maximale Produktionsdauer, $t_{00}^{max}$	108 s	
Maximale Produktionsdauer, $t_{01}^{max}$	276 s	
Maximale Produktionsdauer, $t_{11}^{max}$	197 s	
Maximale Produktionsdauer, $t_{12}^{max}$	219 s	
Maximale Produktionsdauer, $t_{22}^{max}$	230 s	
Mittlere Produktionsdauer, $t_{00}^{\mu}$	72 s	
Mittlere Produktionsdauer, $t_{01}^{\mu}$	233 s	
Mittlere Produktionsdauer, $t_{11}^{\mu}$	197 s	
Mittlere Produktionsdauer, $t_{12}^{\mu}$	219 s	
Mittlere Produktionsdauer, $t_{22}^{\mu}$	230 s	
Anzahl Routen, $AR_{00}$	1	
Anzahl Routen, $AR_{01}$	1	
Anzahl Routen, $AR_{11}$	1	
Anzahl Routen, $AR_{12}$	1	
Anzahl Routen, $AR_{22}$	1	
Anzahl Maschinen, $ M $	3	
Prozessfolgen (Typ 00)	Slave5_M5_3	Fehlerhaft erkannt, da Werkstücke Maschine passieren, aber nicht bearbeitet werden
Prozessfolgen (Typ 01)	Slave5_M5_3	Erste Bearbeitung
	Slave4_M4_2	Zweite Bearbeitung
	Slave3_M3_1	Dritte Bearbeitung
Prozessfolgen (Typ 11)	Slave3_M3_1	Erste Bearbeitung
	Slave4_M4_2	Zweite Bearbeitung
	Slave5_M5_3	Dritte Bearbeitung
Prozessfolgen (Typ 12)	Slave3_M3_1	Erste Bearbeitung
	Slave4_M4_2	Zweite Bearbeitung

Eigenschaft	Wert	Kommentar
	Slave5_M5_3	Dritte Bearbeitung
Prozessfolgen (Typ 22)	Slave3_M3_1	Erste Bearbeitung
	Slave4_M4_2	Zweite Bearbeitung
	Slave5_M5_3	Dritte Bearbeitung
Operationen, $\Omega^{00}$	Slave5_M5_3	Fehlerhaft erkannt, da Werkstücke Maschine passieren, aber nicht bearbeitet werden
Operationen, $\Omega^{01}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{11}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{12}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Operationen, $\Omega^{22}$	{Slave3_M3_1, Slave4_M4_2, Slave5_M5_3}	
Gewichtung der Operation durch M1, $g_{M1}$	6	
Gewichtung der Operation durch M2, $g_{M2}$	6	
Gewichtung der Operation durch M3, $g_{M3}$	10	Eigentlich 6 (jedoch haben 10 Werkstücke die Maschine passiert), erkennbar an Anzahl, die Aktiv war
Anzahl Produkttypen	5	
Häufigkeit Zuordnung, $p_{M1,00}$	0%	
Häufigkeit Zuordnung, $p_{M2,00}$	0%	
Häufigkeit Zuordnung, $p_{M3,00}$	100%	
Häufigkeit Zuordnung, $p_{M1,01}$	100%	
Häufigkeit Zuordnung, $p_{M2,01}$	100%	
Häufigkeit Zuordnung, $p_{M3,01}$	100%	
Häufigkeit Zuordnung, $p_{M1,11}$	100%	
Häufigkeit Zuordnung, $p_{M2,11}$	100%	
Häufigkeit Zuordnung, $p_{M3,11}$	100%	
Häufigkeit Zuordnung, $p_{M1,12}$	100%	
Häufigkeit Zuordnung, $p_{M2,12}$	100%	
Häufigkeit Zuordnung, $p_{M3,12}$	100%	
Häufigkeit Zuordnung, $p_{M1,22}$	100%	
Häufigkeit Zuordnung, $p_{M2,22}$	100%	
Häufigkeit Zuordnung, $p_{M3,22}$	100%	

## Literaturverzeichnis

Die im Rest der Arbeit enthaltenen Verzeichnisse enthalten die zur Erstellung dieser Arbeit herangezogenen Quellen. Diejenigen Quellen, die allgemeiner Literatur i.F.v. Fachbüchern, Fachzeitschriftartikeln und weiteren, nicht vom Autor stammenden, wissenschaftlichen Abhandlungen entstammen, sind im folgenden Literaturverzeichnis enthalten. Sie sind mittels [<Kurzbeleg>] gekennzeichnet. Das darauf folgende Verzeichnis enthält die referenzierten Normen, Richtlinien und Empfehlungen verschiedener Fach-Organisationen und Vereine. Die dort enthaltenen Quellen sind mittels [<Typ><Nummer>] gekennzeichnet. Die im Anschluss dargelegten Internetquellen sind mittels [<Kurzbeleg><sup>®</sup>] gekennzeichnet. Veröffentlichungen des Verfassers befinden sich im darauf folgenden Verzeichnis und sind mittels [<Kurzbeleg><sup>#</sup>] gekennzeichnet. Das letzte Verzeichnis enthält vom Verfasser betreute studentische Arbeiten. Diese sind mittels [<Kurzbeleg><sup>%</sup>] gekennzeichnet.

- [AAC07] E. Amir, M.L. Anderson, V.K. Chaudhi: *Report on DARPA Workshop on Self-Aware Computer Systems*, 2007.
- [ABE90] D. Abel: *Petri-Netze für Ingenieure: Modellbildung und Analyse diskret gesteuerter Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990.
- [ABL02] G. Ammons, R. Bodík, J.R. Larus: *Mining Specifications*. In: *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. New York, USA: ACM (POPL), S. 4–16, 2002.
- [ACH+10] C. Ackermann, R. Cleaveland, S. Huang, A. Ray, C. Shelton, E. Latronico: *Automatic Requirement Extraction from Test Cases*. In: Barringer, Falcone et al. (Hrsg.): *Runtime verification*. Berlin: Springer (Lecture Notes in Computer Science, 6418), 2010.
- [AGM+90] K. Ahrens, E. Götz, J. Möbus, R. Müfler-Baku, G. Schmiedgen: *Produktionsleittechnik: Divergierende Begriffe in Verfahrenstechnik und Fertigungstechnik*. atp edition - Automatisierungstechnische Praxis, Vol. 32 (10), 1990, S. 495–499.
- [AIK+08] D. Arnold, H. Isermann et al. (Hrsg.): *Handbuch Logistik*. 3., neu bearb. Aufl. Berlin: Springer (VDI-Buch), 2008.
- [AL 11] A. Al Khateeb: *Beitrag zur Ermittlung der Gesamtverfügbarkeit in Fließproduktionssystemen*. In: *Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik (WGTL)*, 2011.
- [ALAU13] C. Aldrich, L. Auret: *Unsupervised process monitoring and fault diagnosis with machine learning methods*. London, New York: Springer, 2013 (Advances in computer vision and pattern recognition).
- [ALEL11] T. Al Geddawy, H. El Maraghy: *A model for co-evolution in manufacturing based on biological analogy*. *International Journal of Production Research*, Vol. 49 (15), 2011, S. 4415–4435.
- [ALL10] L.V. Allen: *Verification and anomaly detection for event-based control of manufacturing*. USA, University of Michigan. Dissertation, 2010.
- [ALT10] L.V. Allen, D.M. Tilbury: *Event-based fault detection of manufacturing cell: Data inconsistencies between academic assumptions and industry practice*. In: *IEEE Conference on Automation Science and Engineering (CASE)*, 2010.
- [ALT11] L.V. Allen, D.M. Tilbury: *Necessary condition for a Petri net model that incorporates resources to produce an event stream from an unknown initial state*. In: *American Control Conference (ACC)*, 2011.

- [ALT12] L.V. Allen, D.M. Tilbury: *Anomaly Detection Using Model Generation for Event-Based Systems Without a Preexisting Formal Model*. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, Vol. 42 (3), 2012, S. 654–668.
- [ARBR72] J. Arlt, K. Brankamp: *Zustands- und Leistungskennzahlen der Fertigung: Kontinuierliche Ermittlung über EDV*. Berlin: Beuth, 1972 (Betriebstechnische Reihe RKW/REFA).
- [ASS97] W. Ahrens, H.-J. Scheurlen, G.-U. Spohr: *Informationsorientierte Leittechnik: Informatikmethoden angewandt auf leittechnische Fragestellungen*. München: Oldenbourg, 1997.
- [AWF04] AWF - Arbeitsgemeinschaften für Vitale Unternehmensentwicklung e.V.: *Praktischer Einsatz von Kennzahlen und Kennzahlensystemen in der Produktion*. – Praxisleitfaden der AWF-Arbeitsgemeinschaft "Kennzahlen und Kennzahlensysteme zur Unternehmensführung und -steuerung" 2. Auflage, 2004.
- [BAFA09] M. Barth, A. Fay: *Modellierungstiefen in der Automatisierungstechnik am Beispiel der Prozessindustrie*. In: *ASIM-Treffen STS/GMMS*, 2009.
- [BAHA09] J. Bauer, E. Hayessen: *100 Produktionskennzahlen*. 1. Aufl. Wiesbaden: Cometis, 2009.
- [BANI93] M. Basseville, I.V. Nikiforov: *Detection of abrupt changes: theory and application*. Englewood Cliffs: Prentice Hall, 1993.
- [BAS188] M. Barad, D. Sipper: *Flexibility in manufacturing systems: definitions and Petri net modelling*. International Journal of Production Research, Vol. 26 (2), 1988, S. 237–248.
- [BAS190] M. Barad, D. Sipper: *Flexibility and types of changes in FMSs: A timed petri-nets assessment of machine flexibility*. The International Journal of Advanced Manufacturing Technology, Vol. 5 (4), 1990, S. 292–306.
- [BAT11] Broderick, J. A., L.V. Allen, D.M. Tilbury: *Anomaly detection without a pre-existing formal model: Application to an industrial manufacturing system*. In: *IEEE Conference on Automation Science and Engineering (CASE)*, 2011.
- [BBO+12] S. Braun, C. Bartelt, M. Obermeier, A. Rausch, B. Vogel-Heuser: *Requirements on Evolution Management of Product Lines in Automation Engineering*. In: *Vienna International Conference on Mathematical Modelling (MATHMOD)*, 2012.
- [BDJ15] Birgit Vogel-Heuser, Daniel Schütz, Jens Folmer: *Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS)*. Mechatronics, Vol. 31, 2015, S. 89–100.
- [BELE99] B. Bertsche, G. Lechner: *Zuverlässigkeit im Maschinenbau: Ermittlung von Bauteil- und System-Zuverlässigkeiten*. 2. Aufl. Berlin: Springer, 1999 (VDI-Buch).
- [BEMO92] J.J. Bernardo, Z. Mohamed: *The measurement and use of operational flexibility in the loading of Flexible Manufacturing Systems*. Measuring Manufacturing Flexibility, Vol. 60 (2), 1992, S. 144–155.
- [BEN01] J. Bengtsson: *Manufacturing flexibility and real options: A review: Productive Systems: Strategy, Control, and Management*. International Journal of Production Economics, Vol. 74 (1–3), 2001, S. 213–224.
- [BÉR13] B. Bérard: *An Introduction to Timed Automata*. In: Seatzu (Hrsg.): *Control of discrete-event systems: Automata and Petri net perspectives*. London: Springer (Lecture notes in control and information sciences, 433), S. 169–187, 2013.
- [BESÄ10] M. Bellgran, K. Säfsten: *Production development: Design and operation of production systems*. London: Springer, 2010.

- [BFS10] S. Bergmann, A. Fiedler, S. Straßburger: Generierung und Integration von Simulationsmodellen unter Verwendung des Core Manufacturing Simulation Data (CMSD) Information Model. In: Integrationsaspekte der Simulation: Technik, Organisation und Personal. Karlsruhe: KIT Scientific Publ, S. 461–468, 2010.
- [BGK+07] M. Broy, E. Geisberger, J. Kazmeier, A. Rudorfer, K. Beetz: *Ein Requirements-Engineering-Referenzmodell*. Informatik-Spektrum, Vol. 30 (3), 2007, S. 127–142.
- [BiFE72] A. Biermann, Feldman, J. A.: *On the Synthesis of Finite-State Machines from Samples of Their Behavior*. IEEE Transactions on Computers, Vol. C-21 (6), 1972, S. 592–597.
- [BIR99] A. Birolini: *Reliability engineering: Theory and practice*. 3. Aufl. Berlin, New York: Springer, 1999.
- [BLD00] B. Bordbar, L. Giacomini, D. J. Holding: *UML and Petri nets for design and analysis of distributed systems*. In: *IEEE International Conference on Control Applications*, 2000.
- [BMP+00] R. Beach, A.P. Muhlemann, D.H.R. Price, A. Paterson, J.A. Sharp: *A review of manufacturing flexibility*. European Journal of Operational Research, Vol. 122 (1), 2000, S. 41–57.
- [BMQ+10] D. Behnen, H. Mersch, C. Quix, D. Schmilz, M. Zhang, K. Fayzullin, C. Brecher, U. Epple, M. Jarke: *Gemeinsamkeiten und Unterschiede in der Modellierung von prozesstechnischen und diskreten Produktionssysteme*. In: *Fachtagung Entwurf Komplexer Automatisierungssysteme (EKA)*, Vol. 11, S. 1–10, 2010.
- [BMZ+05] J. Buckley, T. Mens, M. Zenger, A. Rashid, G. Kniesel: *Towards a taxonomy of software change*. Journal of Software Maintenance and Evolution, Vol. 17 (5), 2005, S. 309–332.
- [BPW10] J. Bacher, A. Pöge, K. Wenzig: *Clusteranalyse*. München: Oldenbourg Wissenschaftsverlag GmbH, 2010.
- [BRA15] O. Brauckmann: *Kennzahlen: Vom Zahlenfriedhof zur Wirtschaftlichkeit*. In: *Smart Production: Wertschöpfung durch Geschäftsmodelle*. Berlin, Heidelberg: Springer Berlin Heidelberg, S. 127–143, 2015.
- [BRMA89] P.H. Brill, M. Mandelbaum: *On measures of flexibility in manufacturing systems*. International Journal of Production Research, Vol. 27 (5), 1989, S. 747–756.
- [BRMA90] P.H. Brill, M. Mandelbaum: *Measurement of adaptivity and flexibility in production systems*. European Journal of Operational Research, Vol. 49 (3), 1990, S. 325–332.
- [BRMA97] D. Braha, O. Maimon: *The design process: properties, paradigms, and structure*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 27 (2), 1997, S. 146–166.
- [BRO08] M.G. Brown: *Keeping score: Using the right metrics to drive world-class performance*. New York: Productivity Press, 2008.
- [BSB+11] L. Bassi, C. Secchi, M. Bonfe, C. Fantuzzi: *A SysML-Based Methodology for Manufacturing Machinery Modeling and Design*. IEEE/ASME Transactions on Mechatronics, Vol. 16 (6), 2011, S. 1049–1062.
- [BSK+14] A. Barthelmey, D. Störkle, B. Kuhlenkötter, J. Deuse: *Cyber Physical Systems for Life Cycle Continuous Technical Documentation of Manufacturing Facilities*. Procedia CIRP - Variety Management in Manufacturing, Vol. 17, 2014, S. 207–211.
- [BSK15] A. Barthelmey, D. Störkle, Kuhlenkötter, B.:Deuse, J.: *Bereitstellung und Nutzung technischer Dokumentation in der smarten Fabrik*. In: *Tagungsband Mechatronik*, 2015.
- [BUE00] D.M. Buede: *The engineering design of systems: Models and methods*. New York, NY:

- Wiley, 2000.
- [BUE97] D.M. Buede: *Developing originating requirements: defining the design decisions*. IEEE Transactions on Aerospace and Electronic Systems, Vol. 33 (2), 1997, S. 596–609.
- [BUSA87] E.S. Buffa, R.K. Sarin: *Modern production/operations management*. 8. Aufl. New York: Wiley, 1987 (Wiley Series in production/operations management).
- [BUVY16] I. Buzhinsky, V. Vyatkin: *Plant Model Inference for Closed-Loop Verification of Control Systems: Initial Explorations*. In: *IEEE International Conference on Industrial Informatics (INDIN)*, 2016.
- [CAB09] M.P. Cabasino: *Diagnosis and identification of discrete event systems using Petri nets*. , University of Cagliari. Dissertation, 2009.
- [CAMA09] J.C. Campos, J. Machado: *Pattern-based Analysis of Automated Production Systems*. In: Bakhtadze, Dolgui (Hrsg.): *13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*. Moscow: Elsevier, S. 972–977, 2009.
- [CATE06] S. Cavalieri, S. Terzi: *Proposal of a performance measurement system for the evaluation of scheduling solutions*. International Journal of Manufacturing Technology and Management, Vol. 8 (Nos.1/2/3), 2006, S. 248–263.
- [CBK12] V. Chandola, A. Banerjee, V. Kumar: *Anomaly Detection for Discrete Sequences: A Survey*. IEEE Transactions on Knowledge and Data Engineering, Vol. 24 (5), 2012, S. 823–839.
- [CCB+11] G. Cândido, A.W. Colombo, J. Barata, F. Jammes: *Service-Oriented Infrastructure to Support the Deployment of Evolvable Production Systems*. IEEE Transactions on Industrial Informatics, Vol. 7 (4), 2011, S. 759–767.
- [CDD+00] M. Chouikha, G. Decknatel, R. Drath, G. Frey, C. Müller, C. Simon, J. Thieme, K. Wolter: *Petri Net-Based Descriptions for Discrete-Continuous Systems*. at - Automatisierungstechnik, Vol. 48 (9), 2000, S. 415.
- [CGP01] E.M. Clarke, O. Grumberg, D.A. Peled: *Model checking*. 3. Aufl. Cambridge, Mass: MIT, 2001.
- [CGS06] M.P. Cabasino, A. Giua, C. Seatzu: *Identification of unbounded Petri nets from their coverability graph*. In: *IEEE Conference on Decision and Control (CDC)*, 2006.
- [CGS07] M. Cabasino, A. Giua, C. Seatzu: *Identification of Petri Nets from Knowledge of Their Language*. Discrete Event Dynamic Systems, Vol. 17 (4), 2007, S. 447–474.
- [CHCH96] I. Chen, C.-H. Chung: *An examination of flexibility measurements and performance of flexible manufacturing systems*. International Journal of Production Research, Vol. 34 (2), 1996, S. 379–394.
- [CHCR90] E.J. Chikofsky, J.H. Cross II: *Reverse engineering and design recovery: a taxonomy*. IEEE Software, Vol. 7 (1), 1990, S. 13–17.
- [CHR06] G. Chryssolouris: *Manufacturing Systems: Theory and Practice*. New York: Springer Science+Business Media Inc, 2006.
- [CHR15] L. Christiansen: *Wissensbasiertes Diagnosekonzept durch Kombination von Anlagenstruktur- und Prozessmodell*. Hamburg, Deutschland, Helmut-Schmidt-Universität/Universität der Bundeswehr. Dissertation, 2015.
- [CHTO92] P. Chandra, M.M. Tombak: *Models for the evaluation of routing and machine flexibility: Measuring Manufacturing Flexibility*. European Journal of Operational Research, Vol. 60 (2), 1992, S. 156–165.

- [CoWo98] J.E. Cook, A.L. Wolf: *Discovering Models of Software Processes from Event-based Data*. ACM Transactions on Software Engineering and Methodology, Vol. 7 (3), 1998, S. 215–249.
- [CuFe10] L.G. Curry, R.M. Feldman: *Manufacturing systems modeling and analysis*. 2nd ed. New York: Springer, 2010.
- [DAAL10] R. David, H. Alla: *Discrete, continuous, and hybrid Petri Nets*. 2nd ed. Berlin, London: Springer, 2010.
- [DAGE92] Y. Dallery, S.B. Gershwin: *Manufacturing flow line systems: a review of models and analytical results*. Queueing Systems, Vol. 12 (1-2), 1992, S. 3–94.
- [DAS03] A.I. Dashchenko: *Manufacturing Technologies for Machines of the Future: 21st Century Technologies Volume I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.
- [DFM+08] M. Dotoli, M.P. Fanti, A.M. Mangini, W. Ukovich: *On-line identification of Petri Nets with unobservable transitions*. In: *International Workshop on Discrete Event Systems (WODES)*, S. 449–454, 2008.
- [DFM07] M. Dotoli, M.P. Fanti, A.M. Mangini: *Real time identification of discrete event systems by Petri nets*. In: *IFAC Workshop on Dependable Control of Discrete Systems*, 2007.
- [DHP+93] F. DiCesare, G. Harhalakis, J.M. Proth, M. Silva, F.B. Vernadat: *Practice of Petri Nets in Manufacturing*. London: Chapman & Hall, 1993.
- [Die00] R. Diestel: *Graphentheorie*. 2., neu bearb. und erw. Aufl. Berlin: Springer, 2000 (Springer-Lehrbuch).
- [DiOl03] H. Dierks, E.-R. Olderog: *Temporale Spezifikationslogiken (Temporal Logics for Specifications)*. at - Automatisierungstechnik, Vol. 51 (2), 2003, S. A1-A4.
- [DLH11] C. Diedrich, A. Lüder, L. Hundt: *Bedeutung der Interoperabilität bei Entwurf und Nutzung von automatisierten Produktionssystemen*. at - Automatisierungstechnik, Vol. 59 (7), 2011, S. 426–438.
- [DMB+10] D. Witsch, M. Ricken, B. Kormann, B. Vogel-Heuser: *PLC-statecharts: An approach to integrate umlstatecharts in open-loop control engineering*. In: *IEEE International Conference on Industrial Informatics (INDIN)*, 2010.
- [Dyc06] H. Dyckhoff: *Produktionstheorie: Grundzüge industrieller Produktionswirtschaft*. 5. Aufl. Berlin, Heidelberg: Springer, 2006.
- [ECG+01] M.D. Ernst, J. Cockrell, W.G. Griswold, D. Notkin: *Dynamically discovering likely program invariants to support program evolution*. IEEE Transactions on Software Engineering, Vol. 27 (2), 2001, S. 99–123.
- [Eck15] K. Eckert: *Funktionaler Anwendungsentwurf verteilter Automatisierungssysteme: Anwendung von Entwurfsmustern in der Automatisierungstechnik*. Hamburg, Deutschland, Helmut-Schmidt-Universität/Universität der Bundeswehr. Dissertation, 2015.
- [EGM+09] I. Epifani, C. Ghezzi, R. Mirandola, G. Tamburrelli: *Model Evolution by Run-time Parameter Adaptation*. In: *IEEE International Conference on Software Engineering*. Washington, DC, USA: IEEE Computer Society, S. 111–121, 2009.
- [Egy03] A. Egyed: *A scenario-driven approach to trace dependency analysis*. IEEE Transactions on Software Engineering, Vol. 29 (2), 2003, S. 116–132.
- [ELL12] A.P. Estrada-Vargas, E. López-Mellado, J.-J. Lesage: *Input–output identification of controlled discrete manufacturing systems*. International Journal of Systems Science, Vol. 45 (3), 2012, S. 456–471.

- [ELL14] A.P. Estrada-Vargas, J.-J. Lesage, E. López-Mellado: *A stepwise method for identification of controlled discrete manufacturing systems*. International Journal of Computer Integrated Manufacturing, Vol. 28 (2), 2014, S. 187–199.
- [ELL15] A.P. Estrada-Vargas, E. Lopez-Mellado, J.-J. Lesage: *A Black-Box Identification Method for Automated Discrete-Event Systems*. IEEE Transactions on Automation Science and Engineering, Vol. 14 (3), 2015, S. 1321-1336.
- [ELM05] H.A. ElMaraghy: *Flexible and reconfigurable manufacturing systems paradigms*. International Journal of Flexible Manufacturing Systems, Vol. 17 (4), 2005, S. 261–276.
- [ELM09] H.A. ElMaraghy: *Changing and Evolving Products and Systems – Models and Enablers*. In: ElMaraghy (Hrsg.): *Changeable and Reconfigurable Manufacturing Systems*: Springer London, S. 25–45, 2009.
- [ELM91] S.E. Elmaghraby: *Manufacturing capacity and its measurement: A critical evaluation*. Operations Research and the Rude Intrusions of the Real World, Vol. 18 (7), 1991, S. 615–627.
- [ELS10] J. Esparza, M. Leucker, M. Schlund: *Learning Workflow Petri Nets*. In: Lilius, Penczek (Hrsg.): *Applications and Theory of Petri Nets*: Springer Berlin Heidelberg (Lecture Notes in Computer Science), Vol. 6128, S. 206–225, 2010.
- [ELWI09] H.A. ElMaraghy, H.-P. Wiendahl: *Changeability – An Introduction*. In: ElMaraghy (Hrsg.): *Changeable and Reconfigurable Manufacturing Systems*: Springer London, S. 3–24, 2009.
- [EPP11] U. Epple: *Merkmale als Grundlage der Interoperabilität technischer Systeme*. at - Automatisierungstechnik, Vol. 59 (7), 2011, S. 440–450.
- [FASE08] M.P. Fanti, C. Seatzu: *Fault diagnosis and identification of discrete event systems using Petri nets*. In: *International Workshop on Discrete Event Systems*, 2008.
- [FAY09] A. Fay: *Effizientes Engineering komplexer Automatisierungssysteme*. In: Schnieder (Hrsg.): *Wird der Verkehr automatisch sicherer?: Beschreibungsmittel, Methoden und Werkzeuge des integrierten Systementwurfs zur Fahrzeug- und Verkehrsautomatisierung*, S. 43–60, 2009.
- [FEH+13] Frank T., Eckert K., T. Hadlich, Fay A., Diedrich C., Vogel-Heuser B.: *Erweiterung des V-Modells® für den Entwurf von verteilten Automatisierungssystemen*. at - Automatisierungstechnik, Vol. 61, 2013, S. 79.
- [FEVO13] S. Feldmann, B. Vogel-Heuser: *Änderungsszenarien in der Automatisierungstechnik – Herausforderungen und interdisziplinäre Auswirkungen*. In: Vogel-Heuser (Hrsg.): *Engineering von der Anforderung bis zum Betrieb*: Kassel University Press, S. 95–120, 2013.
- [FKV14] S. Feldmann, K. Kernschmidt, B. Vogel-Heuser: *Combining a SysML-based Modeling Approach and Semantic Technologies for Analyzing Change Influences in Manufacturing Plant Models*. Procedia 5CIRP6, Vol. 17, 2014, S. 451–456.
- [FKV14] S. Feldmann, K. Kernschmidt, B. Vogel-Heuser: *Combining a SysML-based Modeling Approach and Semantic Technologies for Analyzing Change Influences in Manufacturing Plant Models*. Procedia 5CIRP6, Vol. 17, 2014, S. 451–456.
- [FoVo13] Folmer J., B. Vogel-Heuser: *Kenngößenermittlung und -analyse für ein zustandsbasiertes Wartungsmanagement*. In: Vogel-Heuser (Hrsg.): *Engineering von der Anforderung bis zum Betrieb*: Kassel University Press, 2013.
- [FRKO15] T. Frank, H. Koop: *Sicherheitsgerichtete Funktionsbausteine modellbasiert testen: besser und kostengünstiger*. atp edition - Automatisierungstechnische Praxis, Vol. 57 (4), 2015,

- S. 20–23.
- [FRLI00] G. Frey, L. Litz: *Formal methods in PLC programming*. In: *IEEE International Conference on Systems, Man, and Cybernetics*, 2000.
- [FTK16] O. Fakhfakh, A. Toguyeni, O. Korbaa: *On-line fault diagnosis of FMS based on flows analysis*. *Journal of Intelligent Manufacturing*, 2016, S. 1–14.
- [GER87] D. Gerwin: *An Agenda For Research on the Flexibility of Manufacturing Processes*. *International Journal of Operations & Production Management*, Vol. 7 (1), 1987, S. 38–49.
- [GES87] Gesellschaft Materialfluss und Fördertechnik: *Verfügbarkeit von Materialfluss-Systemen*. Düsseldorf: VDI-Verl, 1987 (VDI-Berichte, 636).
- [GJL04] O. Grinchtein, B. Jonsson, M. Leucker: *Learning of event-recording automata*. In: *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*: Springer, S. 379–395, 2004.
- [GLI05] M. Glinz: *Rethinking the Notion of Non-Functional Requirements*. In: *in Proceedings of the Third World Congress for Software Quality (3WCSQ'05)*, S. 55–64, 2005.
- [GLI07] M. Glinz: *On Non-Functional Requirements*. In: *IEEE International Requirements Engineering Conference*, 2007.
- [GMS+03] R.E. Giachetti, L.D. Martinez, O.A. Saenz, C.-S. Chen: *Analysis of the structural measures of flexibility and agility using a measurement theoretical framework*. *International Journal of Production Economics*, Vol. 86 (1), 2003, S. 47–62.
- [GÖFA12A] M. Göring, A. Fay: *Automation Systems – Formal Modeling of Temporal Change of Physical Structure*. In: *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 2012.
- [GÖFA12B] M. Göring, A. Fay: *Beschreibungsmittel für Abhängigkeiten zwischen physikalischen und funktionalen Strukturen*. In: *Fachtagung Entwurf Komplexer Automatisierungssysteme (EKA)*, 2012.
- [GOGEO8] M.W. Godfrey, D.M. German: *The past, present, and future of software evolution*. In: *Frontiers of Software Maintenance (FoSM)*, 2008.
- [GOL78] E.M. Gold: *Complexity of automaton identification from given data*. *Information and Control*, Vol. 37 (3), 1978, S. 302–320.
- [GÖR13] M. Göring: *Description language for the modelling and analysis of temporal change of instrumentation and control system structures*. Hamburg, Deutschland, Helmut-Schmidt-Universität/Universität der Bundeswehr. Dissertation, 2013.
- [GRA11] M. Grande: *100 Minuten für Anforderungsmanagement: Kompaktes Wissen nicht nur für Projektleiter und Entwickler*. 1. Aufl. Wiesbaden: Vieweg + Teubner, 2011.
- [GRME03] M. Grüninger, C. Menzel: *The process specification language (PSL) theory and applications*. *AI Mag*, Vol. 24 (3), 2003, S. 63–74.
- [GRO07] M.P. Groover: *Automation, Production Systems, and Computer-Integrated Manufacturing*. 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.
- [GRO14] M.P. Groover: *Automation, production systems, and computer-integrated manufacturing*. 3. ed., (Pearson internat. ed.). Upper Saddle River, N.J.: Pearson/Prentice Hall, 2014.
- [GUGO89] Y.P. Gupta, S. Goyal: *Flexibility of manufacturing systems: Concepts and measurements*. *European Journal of Operational Research*, Vol. 43 (2), 1989, S. 119–135.
- [GÜT13] K. Güttel: *Konzept zur Generierung von Steuerungscode für Fertigungsanlagen unter Ver-*

- wendung wissensbasierter Methoden. Hamburg, Deutschland, Helmut-Schmidt-Universität/Universität der Bundeswehr. Dissertation, 2013.
- [HAD15] T. Hadlich: *Verwendung von Merkmalen im Engineering von Systemen*. Deutschland, Universität Magdeburg. Dissertation, 2015.
- [HAH+06] J. Huselius, J. Andersson, H. Hansson, S. Punnekkat: *Automatic Generation and Validation of Models of Legacy Software*. In: *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2006.
- [HAU06] T. Hauff: *Prozessleitsysteme: Lebenszyklus und Qualität*. atp edition - Automatisierungstechnische Praxis, Vol. 48 (02), 2006, S. 34–42.
- [HCL+14] G. Hackenberg, A. Campetelli, C. Legat, J. Mund, S. Teufi, B. Vogel-Heuser: *Formal Technical Process Specification and Verification for Automated Production Systems*. In: Amyot, Fonseca i Casas, Pau, Mussbacher (Hrsg.): *System Analysis and Modeling: Models and Reusability*: Springer International Publishing (Lecture Notes in Computer Science), Vol. 8769, S. 287–303, 2014.
- [HEE05] M. Heeg: *Ein Beitrag zur Modellierung von Merkmalen im Umfeld der Prozessleittechnik*. Deutschland, RWTH Aachen. Dissertation, 2005.
- [HLW06] H. Hu, Z. Li, A. Wang: *Mining of Flexible Manufacturing System Using Work Event Logs and Petri Nets*. In: Li, Zaïane, Li (Hrsg.): *Advanced Data Mining and Applications*: Springer Berlin Heidelberg (Lecture Notes in Computer Science), Vol. 4093, S. 380–387, 2006.
- [HNS03] H. Hungar, O. Niese, B. Steffen: *Domain-Specific Optimization in Automata Learning*. In: Hunt, Warren A., Jr., Somenzi (Hrsg.): *Computer Aided Verification*: Springer Berlin Heidelberg (Lecture Notes in Computer Science), Vol. 2725, S. 315–327, 2003.
- [HOF+14] T. Holm, M. Obst, A. Fay, L. Urbas, T. Albers, S. Kreft, U. Hempen: *Dezentrale Intelligenz für modulare Automation: Lösungsansätze für die Realisierung modularer Anlagen*. atp edition - Automatisierungstechnische Praxis (11), 2014, S. 34–43.
- [HOI93] H.-J. Hoitsch: *Produktionswirtschaft: Grundlagen einer industriellen Betriebswirtschaftslehre*. 2. Aufl. München: Vahlen, 1993.
- [HUAN05] J. Huselius, J. Andersson: *Model synthesis for real-time systems*. In: *Ninth European Conference on Software Maintenance and Reengineering*, S. 52–60, 2005.
- [HUMA99] G.Q. Huang, K.L. Mak: *Current practices of engineering change management in UK manufacturing industries*. *International Journal of Operations & Production Management*, Vol. 19 (1), 1999, S. 21–37.
- [HWB+13] C. Haubeck, I. Wior, L. Braubach, A. Pokahr, J. Ladiges, A. Fay, W. Lamersdorf: *Keeping Pace with Changes - Towards Supporting Continuous Improvements and Extensive Updates in Production Automation Software*. *Electronic Communications of the EASST (Volume 56)*, 2013.
- [HWO+10] R. Hametner, D. Winkler, T. Östreicher, S. Biffli, A. Zoitl: *The adaptation of test-driven software processes to industrial automation engineering*. In: *IEEE International Conference on Industrial Informatics (INDIN)*, 2010.
- [ISE06] R. Isermann: *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.
- [JAPe99] M. Jackson, P. Petersson: *Productivity: An Overall Measure of Competitiveness*. In: *CIRP International Seminar on Manufacturing Systems*, 1999.

- [JFW+11] T. Jäger, A. Fay, T. Wagner, U. Löwen: *Mining technical dependencies throughout engineering process knowledge*. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, S. 1–7, 2011.
- [JMR04] J. Aurich, M. Rößing, R. Jaime: *Änderungsmanagement in der Produktion: Am Beispiel der Einführung einer Maschinen- und Betriebsdatenerfassung*. *zwf - Zeitschrift für wirtschaftlichen Fabrikbetrieb* (07-08), 2004, S. 381–384.
- [KFL02] S. Klein, G. Frey, L. Litz: *A petri net based approach to the development of correct logic controllers*. In: *International Workshop on Integration of Specification Techniques for Applications in Engineering (INT)*, 2002.
- [KKP+99] E. Kamsties, A. von Knethen, J. Philipps, B. Schätz: Eine vergleichende Fallstudie mit CASE-Werkzeugen für formale und semi-formale Beschreibungstechniken. Tagungsband des GI/ITG-Fachgespräch Formale Beschreibungstechniken für verteilte Systeme, Vol. 9, 1999, S. 103–112.
- [KLE09] S. Kleuker: *Formale Modelle der Softwareentwicklung: Model-Checking, Verifikation, Analyse und Simulation*. 1. Aufl. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, 2009.
- [KLL05] S. Klein, L. Litz, J.-J. Lesage: *Fault detection of discrete event systems using an identification approach*. In: *IFAC World Congress*, 2005.
- [KMO+10] G. Karsai, F. Massacci, L.J. Osterweil, I. Schieferdecker: *Evolving Embedded Systems*. *Computer*, Vol. 43 (5), 2010, S. 34–40.
- [KoKA98] R. Koller, N. Kastrup: *Prinziplösungen zur Konstruktion technischer Produkte*. 2., neubearbeitete Auflage. Berlin, Heidelberg, s.l.: Springer Berlin Heidelberg, 1998.
- [KOL09] J. Kolditz: *Fachkonzeption für selbststeuernde logistische Prozesse*. Deutschland, Universität Bremen. Dissertation, 2009.
- [KONA92] V.P. Kochikar, T.T. Narendran: *A framework for assessing the flexibility of manufacturing systems*. *International Journal of Production Research*, Vol. 30 (12), 1992, S. 2873–2895.
- [KRA12] J. Krause: *Testfallgenerierung aus modellbasierten Systemspezifikationen auf der Basis von Petrinetzentfaltungen*. 1. Aufl. Herzogenrath: Shaker, 2012.
- [KRAR09] H.-U. Krause, D. Arora: *Controlling-Kennzahlen: Zweisprachiges Handbuch deutsch/englisch = Key performance indicators*. München: Oldenbourg, 2009.
- [KUTE93] H. Kuhn, H. Tempelmeier: *Flexible Fertigungssysteme: Entscheidungsunterstützung für Konfiguration und Betrieb ; mit 98 Tabellen*. Berlin u.a: Springer, 1993.
- [LAD+11] S.-W. Lin, É. André, J.S. Dong, J. Sun, Y. Liu: *An efficient algorithm for learning event-recording automata*. In: *Automated Technology for Verification and Analysis: Springer (Lecture Notes in Computer Science)*, S. 463–472, 2011.
- [LAN99] T. Langemann: *Modellierung als Kernfunktion einer systemorientierten Analyse und Bewertung der diskreten Produktion*. Paderborn, Universität Paderborn, Heinz Nixdorf Institut. Dissertation, 1999.
- [LBK+12] F. Li, G. Bayrak, K. Kernschmidt, B. Vogel-Heuser: *Specification of the Requirements to Support Information Technology-Cycles in the Machine and Plant Manufacturing Industry*. In: *IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, 2012.
- [LBL+14] M. Lochau, J. Bürdek, S. Lity, M. Hagner, C. Legat, U. Goltz, A. Schürr: *Applying Model-based Software Product Line Testing Approaches to the Automation Engineering Domain*.

- at - Automatisierungstechnik, Vol. 62 (11), 2014.
- [LEH78] M.M. Lehman: *Laws of Program Evolution: Rules and Tools for Programming Management*. In: *Proceedings of the Infotech State of the Art Conference: Why Software Projects Fail?*, 1978.
- [LEH80] M. Lehman: *Programs, life cycles, and laws of software evolution*. Proceedings of the IEEE, Vol. 68 (9), 1980, S. 1060–1076.
- [LELE11] D. Lefebvre, E. Leclercq: *Stochastic Petri Net Identification for the Fault Detection and Isolation of Discrete Event Systems*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 41 (2), 2011, S. 213–225.
- [LERA02] M. Lehman, J. Ramil: *Software Evolution and Software Evolution Processes*. Annals of Software Engineering, Vol. 14 (1-4), 2002, S. 275–309.
- [LFV13] C. Legat, J. Folmer, B. Vogel-Heuser: *Evolution in industrial plant automation: A case study*. In: *Conference of the IEEE Industrial Electronics Society (IECON)*, 2013.
- [LIB10] J. Lange, F. Iwanitz, T.J. Burke: *OPC: Von Data Access bis Unified Architecture*. 4. Aufl. Berlin: VDE-Verlag, 2010.
- [LIE99] J. Liebowitz: *Knowledge management handbook*. Boca Raton, Fla: CRC Press, 1999.
- [LIEL07] D. Liekenbrock, J. Elger: *Modellbasiertes Requirements Engineering*. In: *Internet der Dinge*. Berlin [u.a.]: Springer, S. 127–155, 2007.
- [LIFR99] L. Litz, G. Frey: *Methoden und Werkzeuge zum industriellen Steuerungsentwurf - Historie, Stand, Ausblick*. at - Automatisierungstechnik, Vol. 47 (4), 1999.
- [LIG09] P. Liggesmeyer: *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. 2. Aufl. s.l: Spektrum Akademischer Verlag, 2009.
- [LIT09] L. Litz: *Grundlagen der Automatisierungstechnik: Regelungssysteme, Steuerungssysteme, Hybride Systeme*. München, Wien: Oldenbourg, 2009.
- [LMC+14] C. Legat, J. Mund, A. Campetelli, G. Hackenberg, J. Folmer, D. Schütz, M. Broy, B. Vogel-Heuser: *Interface Behavior Modeling for Automatic Verification of Industrial Automation Systems' Functional Conformance*. at - Automatisierungstechnik, Vol. 62 (11), 2014.
- [LMP06] D. Lorenzoli, L. Mariani, M. Pezzè: *Inferring State-based Behavior Models*. In: *International Workshop on Dynamic Systems Analysis (WODA)*, 2006.
- [LRW+97] M.M. Lehman, J.F. Ramil, P.D. Wernick, D.E. Perry, W.M. Turski: *Metrics and Laws of Software Evolution - The Nineties View*. In: *International Software Metrics Symposium*, 1997.
- [LSF+14] C. Legat, F. Steden, S. Feldmann, M. Weyrich, B. Vogel-Heuser: *Co-evolution and reuse of automation control and simulation software: Identification and definition of modification actions and strategies*. In: *Conference of the IEEE Industrial Electronics Society (IECON)*, 2014.
- [LSP+15] H.-Y. Lin, S. Sierla, N. Papakonstantinou, A. Shalyto, V. Vyatkin: *Change Request Management in Model-Driven Engineering of Industrial Automation Software*. In: *IEEE International Conference on Industrial Informatics*, 2015.
- [LUN08] J. Lunze: *Automatisierungstechnik: Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*. 2., überarb. Aufl. München: Oldenbourg, 2008.
- [LUN09] J. Lunze: *Ereignisdiskrete Systeme: Modellierung und Analyse dynamischer Systeme mit Automaten, Markovketten und Petrinetzen ; mit 80 Anwendungsbeispielen und 110*

- Übungsaufgaben*. München, Wien: Oldenbourg, 2009.
- [LWM+12] S. Langer, J. Wilberg, A. Maier, U. Lindemann: *Änderungsmanagement-Report 2012: Studienergebnisse zu Ursachen und Auswirkungen, aktuellen Praktiken, Herausforderungen und Strategien in Deutschland*, 2012.
- [LYCR91] R.L. Lynch, K.F. Cross: *Measure Up!: The Essential Guide to Measuring Business Performance: Mandarin*, 1991.
- [MAN04] S. Manz: *Entwicklung hybrider Komponentenmodelle zur Prozessüberwachung komplexer dynamischer Systeme*. Deutschland, Universität Stuttgart. Dissertation, 2004.
- [MAN15] A. Maier, O. Niggemann: *On the Learning of Timing Behavior for Anomaly Detection in Cyber-Physical Production Systems*. In: *International Workshop on the Principles of Diagnosis (DX)*. Paris, France, 2015.
- [MBS+11] H. Mersch, D. Behnen, D. Schmitz, U. Epple, C. Brecher, M. Jarke: *Gemeinsamkeiten und Unterschiede der Prozess- und Fertigungstechnik*. at - Automatisierungstechnik, Vol. 59 (1), 2011.
- [MCC05] M. Bonfe, C. Fantuzzi, C. Secchi: *Unified Modeling and Verification of Logic Controllers for Physical Systems*. In: *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005.
- [McC95] I. McCarthy: *Manufacturing classification: Lessons from organizational systematics and biological taxonomy*. *Integrated Manufacturing Systems*, Vol. 6 (6), 1995, S. 37–48.
- [MDL+06] J. Machado, B. Denis, J.-J. Lesage, J.-M. Faure, J. C.L. Ferreira Da Silva: *Logic Controllers Dependability Verification Using a Plant Model*. *IFAC Proceedings*, Vol. 39 (17), 2006, S. 37–42.
- [MEEP09] M. Mertens, U. Epple: *Plant asset management functions driven by property models*. In: *IEEE International Conference on Emerging Technologies & Factory Automation (ETFA)*, 2009.
- [MEL005] M.E. Meda-Campana, E. López-Mellado: *Identification of concurrent discrete event systems using Petri nets*. In: *IMACS World Congress on Computational and Applied Mathematics*, S. 11–15, 2005.
- [MEPA02] A. Meyna, B. Pauli: *Taschenbuch der Zuverlässigkeits- und Sicherheitstechnik: Quantitative Bewertungsverfahren*. 1. Aufl. München Wien: Carl Hanser Verlag, 2002.
- [MiMö08] J. Milberg, N. Möller: *Valuation of changeable production systems*. *Production Engineering*, Vol. 2 (4), 2008, S. 417–424.
- [MJG11] C.R. Maga, N. Jazdi, P. Gohner: *Requirements on engineering tools for increasing reuse in industrial automation*. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2011.
- [MJG11A] C. Maga, N. Jazdi, P. Gohner: *Reusable Models in Industrial Automation: Experiences in Defining Appropriate Levels of Granularity*. In: *World Congress of the International Federation of Automatic Control (IFAC)*, 2011.
- [MJG11B] C.R. Maga, N. Jazdi, P. Gohner: *Requirements on engineering tools for increasing reuse in industrial automation*. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2011.
- [MKD15] S. Magnus, J. Krause, C. Diedrich: *Automatisierte Modellgenerierung auf Basis formalisierter Anforderungsbeschreibung*. at - Automatisierungstechnik, Vol. 63 (2), 2015.

- [MMW98] P. Morris, M. Masera, M. Wilkens: *Requirements engineering and industrial uptake*. Requirements Engineering, Vol. 3 (2), 1998, S. 79–83.
- [MoGu96] Moore, K. E., Gupta, S. M.: *Petri net models of flexible and automated manufacturing systems: a survey*. International Journal of Production Research, Vol. 34 (11), 1996, S. 3001–3035.
- [MPP11] L. Mariani, F. Pastore, M. Pezze: *Dynamic Analysis for Diagnosing Integration Faults*. Software Engineering, IEEE Transactions on, Vol. 37 (4), 2011, S. 486–508.
- [MUK+02] M.G. Mehrabi, A.G. Ulsoy, Y. Koren, P. Heytler: *Trends and perspectives in flexible and reconfigurable manufacturing systems*. Journal of Intelligent Manufacturing, Vol. 13 (2), 2002, S. 135–146.
- [MUK00] M.G. Mehrabi, A.G. Ulsoy, Y. Koren: *Reconfigurable manufacturing systems: Key to future manufacturing*. Journal of Intelligent Manufacturing, Vol. 11 (4), 2000, S. 403–419.
- [MUR89] T. Murata: *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, Vol. 77 (4), 1989, S. 541–580.
- [NA 35] NA 35. *Abwicklung von PLT-Projekten*, 2003.
- [NAG92] N. Nagarur: *Some performance measures of flexible manufacturing systems*. International Journal of Production Research, Vol. 30 (4), 1992, S. 799–809.
- [NAH93] S. Nahmias: *Production and operations analysis*. 2. ed. Homewood, Ill.: Irwin, 1993.
- [NER58] A. Nerode: *Linear automaton transformations*. Proceedings of the American Mathematical Society, Vol. 9 (4), 1958, S. 541.
- [NGP95] A. Neely, M. Gregory, K. Platts: *Performance measurement system design: a literature review and research agenda*. International Journal of Operations & Production Management, Vol. 15 (4), 1995, S. 80–116.
- [NHR+08] P. Nyhuis, T. Heinen, G. Reinhart, C. Rimpau, E. Abele, A. Wörn: *Wandlungsfähige Produktionssysteme: Theoretischer Hintergrund zur Wandlungsfähigkeit von Produktionssystemen*, Vol. 98 (H. 1/2), 2008, S. 85–91.
- [NIE09] J. Niemann: *Life Cycle Management*. In: Bullinger, Spath et al. (Hrsg.): *Handbuch Unternehmensorganisation: Strategien, Planung, Umsetzung*. Berlin: Springer, 2009.
- [NIFR15] O. Niggemann, C. Frey: *Data-driven anomaly detection in cyber-physical production systems*. at - Automatisierungstechnik, Vol. 63 (10), 2015.
- [NIG14] O. Niggemann: *Industrie 4.0 ohne modellbasierte Softwareentwicklung*. atp edition - Automatisierungstechnische Praxis, Vol. 56 (5), 2014, S. 22–30.
- [NlLo15] O. Niggemann, V. Lohweg: *On the Diagnosis of Cyber-Physical Production Systems: State-of-the-Art and Research Agenda*. In: *Conference on Artificial Intelligence (AAAI)*, 2015.
- [NMP+00] A. Neely, J. Mills, K. Platts, H. Richards, M. Gregory, M. Bourne, M. Kennerley: *Performance measurement system design: developing and testing a process-based approach*. International Journal of Operations & Production Management, Vol. 20 (10), 2000, S. 1119–1145.
- [NMS+13] P. Novák, M. Melik-Merkumians, M. Steinegger, T. Moser, R. Šindelář, A. Zoitl: *Extraction of Automation System Engineering Knowledge for Mapping Plant and Simulation Interfaces*. In: Borangiu, Thomas, Trentesaux (Hrsg.): *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*. Berlin, Heidelberg: Springer, S. 247–261, 2013.
- [NWW97] S.Y. Nof, W.E. Wilhelm, H.J. Warnecke: *Industrial Assembly*. Boston, MA: Springer US,

- 1997.
- [NYS+00] R. Narain, R.C. Yadav, J. Sarkis, J.J. Cordeiro: *The strategic implications of flexibility in manufacturing systems*. International Journal of Agile Management Systems, Vol. 2 (3), 2000, S. 202–213.
- [OBA+12] O. Niggemann, B. Stein, A. Vodencarevic, A. Maier, H. Kleine Büning: *Learning Behavior Models for Hybrid Timed Systems*. In: *Conference on Artificial Intelligence (AAAI)*, 2012.
- [OBJ12] Object Management Group: *OMG Systems Modeling Language (OMG SysML™): Version 1.3 with change bars*. <http://sysml.org/docs/specs>, 2012.
- [OES09] B. Oestereich: *Analyse und Design mit UML 2.3: Objektorientierte Softwareentwicklung*. 9. aktualisierte Auflage. München: Oldenbourg Wissenschaftsverlag GmbH, 2009.
- [OPH05] L. Ophey: *Entwicklungsmanagement: Methoden in der Produktentwicklung*. Berlin: Springer, 2005.
- [OPP02] M.S. Obaidat, G.I. Papadimitriou, A.S. Pomportsis: Guest editorial learning automata: theory, paradigms, and applications. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society, Vol. 32 (6), 2002, S. 706–709.
- [ORG+11] B. Opgenoorth, J.H. Richter, T. Grosch, D. Wolff, A. Fay: *Verlässlichkeitsanforderungen in der Prozess- und Ressourcenbeschreibung: Kombination von VDI/VDE 3682 und IEC 62424 (CAEX)*. atp edition - Automatisierungstechnische Praxis, Vol. 53 (3), 2011, S. 9.
- [PAW199] R.P. Parker, A. Wirth: *Manufacturing flexibility: Measures and relationships*. European Journal of Operational Research, Vol. 118 (3), 1999, S. 429–449.
- [PEFA13] M. Perin, J.-M. Faure: *Building meaningful timed models of closed-loop DES for verification purposes*. Control Engineering Practice, Vol. 21 (11), 2013, S. 1620–1639.
- [PET62] C.A. Petri: *Kommunikation mit Automaten*. Bonn, Universität Bonn, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik. Dissertation, 1962.
- [PIGO04] S.S. Pinter, M. Golani: *Discovering workflow models from activities' lifespans*. Computers in Industry, Vol. 53 (3), 2004, S. 283–296.
- [PoFE91] S. Porat, J. Feldman: *Learning Automata from Ordered Examples*. Machine Learning, Vol. 7 (2-3), 1991, S. 109–138.
- [PORU11] K. Pohl, C. Rupp: *Basiswissen Requirements Engineering: Aus- und Weiterbildung zum "Certified Professional for Requirements Engineering" ; Foundation Level nach IREB-Standard*. 3. Aufl. Heidelberg: dpunkt-Verl, 2011.
- [PSG14] H. Prähofer, R. Schatz, A. Grimmer: *Behavioral Model Synthesis of PLC Programs from Execution Traces*. In: *IEEE International Conference on Emerging Technology and Factory Automation (ETFA)*, 2014.
- [RBC+10] L. Ribeiro, J. Barata, G. Cândido, M. Onori: *Evolvable Production Systems: An Integrated View on Recent Developments*. In: *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*, 2010.
- [REF84] REFA - Verband für Arbeitsstudien und Betriebsorganisation: *Methodenlehre des Arbeitsstudiums: Teil 1: Grundlagen*. 7. Aufl. München: Hanser, 1984.
- [REF90] REFA - Verband für Arbeitsstudien und Betriebsorganisation: *Planung und Gestaltung komplexer Produktionssysteme*. 2. Aufl. München: Hanser, 1990.

- [RFT15] L. Racchetti, C. Fantuzzi, L. Tacconi: Verification and validation based on the generation of testing sequences from timing diagram specifications in industrial automation. In: Conference of the IEEE Industrial Electronics Society (IECON), S. 2816–2821, 2015.
- [RLL09A] M. Roth, J.-J. Lesage, L. Litz: *Distributed identification of concurrent discrete event systems for fault detection purposes*. In: *European Control Conference (ECC)*, 2009.
- [RLL09B] M. Roth, J.-J. Lesage, L. Litz: *The concept of residuals for fault localization in discrete event systems*. In: *IFAC Workshop on Dependable Control of Discrete Systems (DCDS)*, 2009.
- [RLL10] M. Roth, J.-J. Lesage, L. Litz: *Black-box identification of discrete event systems with optimal partitioning of concurrent subsystems*. In: *American Control Conference (ACC)*, 2010.
- [RLL98] D. Rowe, J. Leaney, D. Lowe: *Defining Systems Evolvability - A Taxonomy of Change*. In: *IEEE International Conference and Workshop: Engineering of Computer-Based Systems*, 1998.
- [ROG11] S. Rogalski: *Flexibility measurement in production systems: Handling uncertainties in industrial production*. Berlin: Springer, 2011.
- [RoKJ09] V. Rogstrand, T. Kjellberg: *The representation of manufacturing requirements in model-driven parts manufacturing*. *International Journal of Computer Integrated Manufacturing*, Vol. 22 (11), 2009, S. 1065–1072.
- [RSL+12] M. Roth, S. Schneider, J.-J. Lesage, L. Litz: *Fault detection and isolation in manufacturing systems with an identified discrete event model*. *International Journal of Systems Science*, Vol. 43 (10), 2012, S. 1826–1841.
- [RTF15] L. Racchetti, L. Tacconi, C. Fantuzzi: Generating automatically the documentation from PLC code by D4T3 to improve the usability and life cycle management of software in automation. In: *IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.
- [RTS+14] S. Rösch, D. Tikhonov, D. Schütz, B. Vogel-Heuser: *Model-based testing of PLC software: test of plants' reliability by using fault injection on component level*. In: *19th IFAC World Congress (IFAC 2014)*. Cape Town, South Africa, S. 3509–3515, 2014.
- [RUP+15] S. Rösch, S. Ulewicz, J. Provost, B. Vogel-Heuser: Review of Model-Based Testing Approaches in Production Automation and Adjacent Domains—Current Challenges and Research Gaps. *Journal of Software Engineering and Applications*, Vol. 08 (09), 2015, S. 499–519.
- [SBK06] H. Shivanad, M. Benal, V. Koti: *Flexible Manufacturing Systems*. New Delhi: New Age International (P) Ltd., 2006.
- [ScFA13] F. Schumacher, A. Fay: *Konzept und Werkzeugunterstützung zur automatischen Generierung von IEC 61131-3 konformen Steuerungsalgorithmen auf Basis einer GRAFCET-Spezifikation*. In: *Tagungsband "Automation"*, 2013.
- [SCGR08] W. Schorn, N. Große: *Hierarchien und Symbolik in der Produktionstechnik*. atp edition-Automatisierungstechnische Praxis, Vol. 50 (11), 2008, S. 66–74.
- [SCH08] R. Schady: *Methode und Anwendungen einer wissensorientierten Fabrikmodellierung*. Magdeburg, Deutschland, Otto-von-Guericke-Universität. Dissertation, 2008.
- [SCH10] M. Schenk: *Instandhaltung technischer Systeme: Methoden und Werkzeuge zur Gewährleistung eines sicheren und wirtschaftlichen Anlagenbetriebs*. Berlin, Heidelberg: Springer-Verlag, 2010.
- [SCH11] H. Schwinn: *Requirements Engineering*. München: Oldenbourg Wissenschaftsverlag, 2011.

- [SCH13] S. Schreiber: *Entwicklung einer Vergleichs- und Bewertungsmöglichkeit von dezentralen Steuerungsarchitekturen für Produktionssysteme*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Dissertation, 2013.
- [SCH16] S. Schröck: *Interdisziplinäre Wiederverwendung im Engineering automatisierter Anlagen: Anforderungen, Konzept und Umsetzung für die Prozessindustrie*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Dissertation, 2016.
- [SCH99] E. Schnieder: *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme*. Wiesbaden: Vieweg+Teubner Verlag, 1999.
- [ScSc10] E. Schnieder, L. Schnieder: *Terminologische Präzisierung des Systembegriffs: Grundlage formaler Systembeschreibungen*. atp edition - Automatisierungstechnische Praxis, Vol. 52 (09), 2010, S. 46–59.
- [ScWi04] M. Schenk, S. Wirth: *Fabrikplanung und Fabrikbetrieb: Methoden für die wandlungsfähige und vernetzte Fabrik*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2004.
- [SDB+83] K. Stecke, D. Dubois, J. Browne, S. Sethi, K. Rathmill: *Classification of flexible manufacturing systems: Evolution towards the automated Factory*. Michigan, USA, The University of Michigan. Working Paper, 1983.
- [SDK+11] J. Stammel, Z. Durdik, K. Krogmann, R. Weiss, H. Koziolk: *ABB DARWIN - Software Evolution for Industrial Automation Systems: Literature Overview*, 2011.
- [SEA13] C. Seatzu (Hrsg.): *Control of discrete-event systems: Automata and Petri net perspectives*. London: Springer (Lecture notes in control and information sciences, 433), 2013.
- [SEA14] C. Seatzu: *Fault diagnosis of discrete event systems using Petri nets*. In: *International Workshop on Verification and Evaluation of Computer and Communication Systems (VECOS)*, 2014.
- [SESE90] A. Sethi, S. Sethi: *Flexibility in manufacturing: A survey*. International Journal of Flexible Manufacturing Systems, Vol. 2 (4), 1990, S. 289–328.
- [SEW14] G. Seebacher, H. Winkler: *Evaluating flexibility in discrete manufacturing based on performance and efficiency*. International Journal of Production Economics, Vol. 153, 2014, S. 340–351.
- [SFL15] J. Saives, G. Faraut, J.-J. Lesage: *Identification of Discrete Event Systems Unobservable Behaviour by Petri nets using Language Projections*. In: *IEEE European Control Conference (ECC)*, 2015.
- [SFT+10] M. Strube, A. Fay, S. Truchat, H. Figalist: *Funktionale Anlagenbeschreibung als Basis der Modernisierungsplanung*. In: *Kongress Automation*, 2010.
- [SiWE92] D. Sinha, J.C. Wei: *Stochastic analysis of flexible process choices*. European Journal of Operational Research, Vol. 60 (2), 1992, S. 183–199.
- [SJF11A] S. Schreiber, S. Jerentz, A. Fay: *Anforderungen an Steuerungskonzepte für moderne Fertigungsanlagen: Herausforderungen für dezentrale Ansätze in der Automatisierungstechnik*. In: *Kongress Automation*, 2011.
- [SJF11B] M. Strube, T. Jäger, A. Fay: *Integriertes Engineering durch Zusammenführen von Prozess- und Anlagenbeschreibung – Ein Konzept zur ganzheitlichen Beschreibung von Produktionssystemen*. In: *IFF Wissenschaftstage*, 2011.

- [SLD11] S. Schneider, L. Litz, M. Danancher: *Timed residuals for fault detection and isolation in discrete event systems*. In: *International Workshop on Dependable Control of Discrete Systems (DCDS)*, 2011.
- [SLL12] S. Schneider, L. Litz, J. Lesage: *Determination of timed transitions in identified discrete-event models for fault detection*. In: *IEEE Annual Conference on Decision and Control (CDC)*, 2012.
- [SPJ+03] H.W. Schmidt, I.D. Peake, Jue Xie, I. Thomas, B.J. Kramer, A. Fay, P. Bort: *Modelling Predictable Component-Based Distributed Control Architectures*. In: *IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS)*, S. 339, 2003.
- [SPK15] J. Stefan, C. Paiz, M. Köster: *Anomalien und Ineffizienz in Produktionssysteme erkennen*. atp edition, Vol. 57 (10), 2015, S. 26–37.
- [SRF+11] M. Strube, S. Runde, H. Figalst, A. Fay: *Risk minimization in modernization projects of plant automation - A knowledge-based approach by means of semantic web technologies*. In: *IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, 2011.
- [STFA11] M. Strube, A. Fay: *Modellgestützte Modernisierungsplanung: Ist-Zustand mit CAEX abbilden*. atp edition - Automatisierungstechnische Praxis, Vol. 53 (7-8), 2011, S. 46–52.
- [SWV10] T. Schmidt, D. Wustmann, V. Vasyutynskyy: *Identifizierung und Bewertung von Anomalien im Materialflussverhalten auf der Basis von Ereignisdaten*. In: *ASIM-Fachtagung „Simulation in Produktion und Logistik“ - Integrationsaspekte der Simulation: Technik, Organisation und Personal*, 2010.
- [T.G.06] T. Hussain, G. Frey: *UML-based Development Process for IEC 61499 with Automatic Test-case Generation*. In: *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2006.
- [TAY97] G.D. Taylor: *Industry Best Practices: A Guide to Benchmarking Manufacturing and Production Operations*. In: *Perspectives on Material Handling*, 1997.
- [TEL11] C. Torens, L. Ebrecht, K. Lemmer: *Inverse Model Based Testing - Generating Behavior Models from Abstract Test Cases*. In: *International Conference on Software Testing, Verification and Validation Workshops*, 2011.
- [TLE+14] T. Tapia-Flores, E. Lopez-Mellado, A.P. Estrada-Vargas, J.-J. Lesage: *Petri net discovery of discrete event processes by computing t-invariants*. In: *IEEE International Conference on Emerging Technology and Factory Automation (ETFA)*, 2014.
- [TOM10] J.A. Tompkins: *Facilities planning*. 4th ed. Hoboken, NJ: Wiley, 2010.
- [ToTo98] A. de Toni, S. Tonchia: *Manufacturing flexibility: a literature review*. *International Journal of Production Research*, Vol. 36 (6), 1998, S. 1587–1617.
- [TSPH98] N. Tsourveloudis, Y. Phillis: *Manufacturing flexibility measurement: a fuzzy logic framework*. *IEEE Transactions on Robotics and Automation*, Vol. 14 (4), 1998, S. 513–524.
- [TTV09] W. Terkaj, T. Tolio, A. Valente: *Focused Flexibility in Production Systems*. In: ElMaraghy (Hrsg.): *Changeable and Reconfigurable Manufacturing Systems*: Springer London, S. 47–66, 2009.
- [TUR09] V. Turau: *Algorithmische Graphentheorie*. 2., überarb. Aufl. München, Wien: Oldenbourg, 2009.
- [TWN98] A. Tharumarajah, A.J. Wells, L. Nemes: *Comparison of emerging manufacturing concepts*. In: *IEEE International Conference on Systems, Man, and Cybernetics*, 1998.

- [ULVO16] S. Ulewicz, B. Vogel-Heuser: *Guided semi-automatic system testing in factory automation*. In: *IEEE International Conference on Industrial Informatics (INDIN)*, S. 142–147, 2016.
- [USV+14] S. Ulewicz, D. Schütz, B. Vogel-Heuser, B. Korajda, D. Hess: *Modellbasierte Auswirkungsbewertung von Änderungen und Testanpassung für SPS-Steuerungssoftware im Maschinen- und Anlagenbau*. In: *Kongress Automation*, 2014.
- [VABE03] W. van der Aalst, E. Best (Hrsg.): *Applications and Theory of Petri Nets 2003: Springer Berlin Heidelberg (Lecture Notes in Computer Science)*, 2003.
- [VAMA12] D. van der Linden, H. Mannaert: *In Search of Rules for Evolvable and Stateful run-time Deployment of Controllers in Industrial Automation Systems*. In: *International Workshop on Computer Vision and Computer Graphics (ICONS)*, 2012.
- [VAN06] W. van der Aalst: *Process mining and monitoring processes and services: Workshop report*. In: *Dagstuhl Seminar*, 2006.
- [VAN11] W. van der Aalst: *Process mining: Discovery, conformance and enhancement of business processes*. Berlin: Springer, 2011.
- [VAN12] W. van der Aalst: *Process Mining*. Communications of the ACM, Vol. 55 (8), 2012, S. 76–83.
- [VAWE04] W. van der Aalst, A.J. Weijters: *Process mining: a research agenda*. Computers in Industry, Vol. 53 (3), 2004, S. 231–244.
- [VDF+13] B. Vogel-Heuser, C. Diedrich, A. Fay, P. Göhner: *Anforderungen an das Software-Engineering in der Automatisierungstechnik*. In: *Software Engineering*, S. 51–66, 2013.
- [VDF+14] B. Vogel-Heuser, C. Diedrich, A. Fay, S. Jeschke, S. Kowalewski, M. Wollschlaeger, P. Göhner: *Challenges for Software Engineering in Automation*. Journal of Software Engineering and Applications, Vol. 07 (05), 2014, S. 440–451.
- [VDMc14] VDMA, McKinsey & Company: *Zukunftsperspektive deutscher Maschinenbau: Erfolgreich in einem dynamischen Umfeld agieren*. – Studie. <http://www.vdma.org/zukunftsperspektive>, 2014.
- [VER10] S.E. Verwer: *Efficient identification of timed automata: theory and practice*. Niederlande, Delft University of Technology. Dissertation, 2010.
- [VFS+15] B. Vogel-Heuser, A. Fay, I. Schäfer, M. Tichy: *Evolution of software in automated production systems: Challenges and Research Directions*. Journal of Systems and Software, Vol. 110, 2015, S. 54–84.
- [VISR11] V. Vinod, R. Sridharan: *Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system*. International Journal of Production Economics, Vol. 129 (1), 2011, S. 127–146.
- [VKN+11] A. Vodencarevic, H. Kleine Büning, O. Niggemann, A. Maier: *Identifying Behavior Models for Process Plants*. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2011.
- [VKT+13] B. Vogel-Heuser, B. Kormann, D. Tikhonov, S. Rösch: *Automatisierter modellbasierter Applikationstest für SPS Steuerungsprogramme auf der Basis von UML*. at - Automatisierungstechnik, Vol. 61 (6), 2013, S. 382–392.
- [VLF+14] B. Vogel-Heuser, C. Legat, J. Folmer, S. Feldmann: *Researching Evolution in Industrial Plant Automation: Scenarios and Documentation of the Pick and Place Unit*. – Technical Report (TUM-AIS-TR-01-14-02). <https://mediatum.ub.tum.de/node?id=1208973>, 2014.
- [VoO'00] R.J. Vokurka, S.W. O'Leary-Kelly: *A review of empirical research on manufacturing flexibility*.

- ty. *Journal of Operations Management*, Vol. 18 (4), 2000, S. 485–501.
- [VRF+16] B. Vogel-Heuser, S. Rösch, J. Fischer, T. Simon, S. Ulewicz, J. Folmer: Fault Handling in PLC-Based Industry 4.0 Automated Production Systems as a Basis for Restart and Self-Configuration and Its Evaluation. *Journal of Software Engineering and Applications*, Vol. 09 (01), 2016, S. 1–43.
- [VRM+15] B. Vogel-Heuser, S. Rösch, A. Martini, M. Tichy: *Technical Debt in Automated Production Systems*. In: *International Conference on Software Maintenance and Evolution*, 2015.
- [VSF+14] B. Vogel-Heuser, D. Schütz, T. Frank, C. Legat: *Model-driven engineering of Manufacturing Automation Software Projects – A SysML-based approach*. *Mechatronics*, Vol. 24 (7), 2014, S. 883–897.
- [VWW08] S.E. Verwer, M.M. de Weerd, C. Witteveen: *Efficiently learning simple timed automata*. In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2008.
- [VYA13] V. Vyatkin: *Software Engineering in Industrial Automation: State-of-the-Art Review*. *IEEE Transactions on Industrial Informatics*, Vol. 9 (3), 2013, S. 1234–1249.
- [WAH05] M.I.M. Wahab: *Measuring machine and product mix flexibilities of a manufacturing system*. *International Journal of Production Research*, Vol. 43 (18), 2005, S. 3773–3786.
- [WAL12] A.-M. Walters: *Veränderungen effizient managen: Verwaltung von Anlagendaten in der Chemieindustrie*. *cav Chemie Anlagen Verfahren* (9), 2012, S. 72–73.
- [WEN+07] H.-P. Wiendahl, H.A. ElMaraghy, P. Nyhuis, M.F. Zäh, H.-H. Wiendahl, N. Duffie, M. Brieke: *Changeable Manufacturing - Classification, Design and Operation*. *CIRP Annals - Manufacturing Technology*, Vol. 56 (2), 2007, S. 783–809.
- [WES03] E. Westkämper: *Adaptable Production Structures*. In: Dashchenko (Hrsg.): *Manufacturing Technologies for Machines of the Future*: Springer Berlin Heidelberg, S. 87–120, 2003.
- [WES06] E. Westkämper: *Einführung in die Organisation der Produktion*. Berlin: Springer, 2006.
- [WEWA10] E. Westkämper, H.-J. Warnecke: *Einführung in die Fertigungstechnik*. Wiesbaden: Vieweg+Teubner, 2010.
- [WHI96] G.P. White: *A survey and taxonomy of strategy-related performance measures for manufacturing*. *International Journal of Operations & Production Management*, Vol. 16 (3), 1996, S. 42–61.
- [WIT12] M. Witte: *System Engineering, Plant Engineering and Functional Models*. *Softwaretechnik - Trends*, Vol. 32 (2), 2012, S. 94–95.
- [WML99] S.T.W. Wong, K.L. Mak, H.Y.K. Lau: *An object-oriented model for the specification of manufacturing systems*. *Computers & Industrial Engineering*, Vol. 36 (3), 1999, S. 655–671.
- [WRG+10] G. Weigert, O. Rose, P. Gocev, G. Mayer: *Kennzahlen zur Bewertung logistischer Systeme*. In: *Fachtagung Simulation in Produktion und Logistik*, 2010.
- [WVS+10] D. Wustmann, V. Vasyutynskyy, T. Schmidt, K. Kabitzsch: *Diagnose und Optimierung von Materialflusststeuerungen. – Schlussbericht des aus dem Haushaltsmitteln des Bundesministeriums für Wirtschaft und Arbeit (BMWA) über die Arbeitsgemeinschaft industrieller Forschungsvereinigungen „Otto von Guericke“ e.V. (AiF) im Auftrag der Bundesvereinigung Logistik (BVL) e.V. geförderten Forschungsvorhaben (AiF-Nr. 15770 BR)*, 2010.
- [WWL08] M. Wahab, D. Wu, C.-G. Lee: *A generic approach to measuring the machine flexibility of manufacturing systems*. *European Journal of Operational Research*, Vol. 186 (1), 2008,

- S. 137–149.
- [XIO11] J. Xiong: *New software engineering paradigm based on complexity science: An introduction to NSE*. New York: Springer, 2011.
- [ZAMU07] M.F. Zaeh, N. Mueller: *A modeling approach for evaluating capacity flexibilities in uncertain markets*. International Journal of Flexible Manufacturing Systems, Vol. 19 (3), 2007, S. 151–172.
- [ZBF11] F. Zammori, M. Braglia, M. Frosolinia: *A measurement method of routing flexibility in manufacturing systems*. International Journal of Industrial Engineering Computations, 2011, S. 593–616.
- [ZIWE06] O. Zirn, S. Weikert: *Modellbildung und Simulation hochdynamischer Fertigungssysteme*. Berlin/Heidelberg: Springer-Verlag, 2006.
- [ZÖR88] G. Zörntlein: *Flexible Fertigungssysteme: Belegung, Steuerung, Datenorganisation*. München, Wien: Carl Hanser Verlag, 1988.
- [ZVE10] ZVEI e.V.: *Life-Cycle-Management für Produkte und Systeme der Automation: Ein Leitfaden des Arbeitskreises Systemaspekte im ZVEI Fachverband Automation*, 2010.

## Referenzierte Normen, Richtlinien und Empfehlungen

- [DIN 8743] DIN 8743. *Verpackungsmaschinen und Verpackungsanlagen – Kennzahlen zur Charakterisierung des Betriebsverhaltens und Bedingungen für deren Ermittlung im Rahmen eines Abnahmelaufs*, 2013.
- [DIN 31051] DIN 31051. *Grundlagen der Instandhaltung*, 2012.
- [DIN 8580] DIN 8580. *Fertigungsverfahren - Begriffe, Einteilung*, 2003.
- [DIN EN 61131-1] DIN EN 61131-1. *Speicherprogrammierbare Steuerungen – Teil 1: Allgemeine Informationen*, 2004.
- [DIN EN 61508-3] DIN EN 61508-3. *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme - Teil 3: Anforderungen an Software*, Februar 2011.
- [DIN EN 61512-1] DIN EN 61512-1. *Chargenorientierte Fahrweise Teil 1: Modelle und Terminologie*, 1999.
- [DIN EN 62264-1] DIN EN 62264-1. *Integration von Unternehmens-EDV und Leitsystemen - Teil 1: Modelle und Terminologie*, 2012.
- [DIN V 19222] DIN V 19222. *Leittechnik - Begriffe*, 2001.
- [IEEE Std 1233] IEEE Std 1233. *IEEE Guide for Developing Requirements Specifications*, 1998.
- [ISO 18629] ISO 18629. *Industrial automation systems and integration - Process specification language*, 2004.
- [ISO 22400-2] ISO 22400-2. *Automation systems and integration - Key performance indicators for manufacturing operations management - Part 2: Definitions and descriptions*, 2012.
- [ISO/IEC 15909] ISO/IEC 15909. *Software and system engineering - High-level Petri nets*, 2004.
- [ISO/IEC 19505] ISO/IEC 19505. *Information technology - Object Management Group Unified Modeling Language (OMG UML)*, 2012.
- [ISO/IEC 25010] ISO/IEC 25010. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*, 2011.
- [ISO/IEC 25020] ISO/IEC 25020. *Systems and software engineering — Systems and software Quality requirements and evaluation (SQuaRE) — Measurement reference model and guide*, 2007.
- [ISO/IEC 25021] ISO/IEC 25021. *Systems and software engineering — Systems and software Quality requirements and evaluation (SQuaRE) — Quality measure elements*, 2012.
- [ISO/IEC 25022] ISO/IEC 25022, FDIS. *Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use*, 2016.
- [ISO/IEC 25023] ISO/IEC 25023, FDIS. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality*, 2016.
- [ISO/IEC 26702] ISO/IEC 26702, IEEE Std 1220-2005. *IEEE Standard for Application and Management of the Systems Engineering Process - Application and Management of the systems engineering process*, 2005.
- [NA 35] NA 35. *Abwicklung von PLT-Projekten*, 2003.

- [VDI 2206] VDI 2206. *Entwicklungsmethodik für mechatronische Systeme*, 2004.
- [VDI 2221] VDI 2221. *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*, 1993.
- [VDI 2525] VDI 2525. *Praxisorientierte Logistikkennzahlen für kleine und mittelständische Unternehmen*, 1999.
- [VDI 2689] VDI 2689. *Leitfaden für Materialflußuntersuchungen*, 1974.
- [VDI 2893] VDI 2893. *Auswahl und Bildung von Kennzahlen für die Instandhaltung*, 2006.
- [VDI 3649] VDI 3649. *Anwendung der Verfügbarkeitsrechnung für Förder- und Lagersysteme*, 1992.
- [VDI 3978] VDI 3978. *Durchsatz und Spielzeiten in Stückgut-Fördersystemen*, 1998.
- [VDI 4400 - 2] VDI 4400 - 2. *Logistikkennzahlen für die Produktion*, 2004.
- [VDI 4486] VDI 4486. *Zuverlässigkeit in der Intralogistik - Leistungsverfügbarkeit*, 2012.
- [VDI 4490] VDI 4490. *Operative Logistikkennzahlen von Wareneingang bis Versand*, 2007.
- [VDI/VDE 3542 - 2] VDI/VDE 3542 - 2. *Sicherheitstechnische Begriffe für Automatisierungssysteme Zuverlässigkeit und Sicherheit komplexer Systeme (Begriffe)*, 2000.
- [VDI/VDE 3681] VDI/VDE 3681. *Einordnung und Bewertung von Beschreibungsmitteln aus der Automatisierungstechnik*, 2005.
- [VDI/VDE 3694] VDI/VDE 3694. *Lastenheft/Pflichtenheft für den Einsatz von Automatisierungssystemen*, 2008.
- [VDMA 66412-1] VDMA 66412-1. *Manufacturing Execution Systems (MES) Kennzahlen*, 2009.

## Referenzierte Internetquellen

- [DFG17<sup>®</sup>] DFG Priority Programme 1593: *Design For Future – Managed Software Evolution*. <http://www.dfg-spp1593.de/> [Stand: 2017] [Abruf: 2017-05-19]
- [NAT02<sup>®</sup>] National Institute of Standards and Technology: *PSL Ontology -- Current Theories and Extensions*. <http://www.mel.nist.gov/psl/psl-ontology/> [Stand: 2002] [Abruf: 2017-05-19]
- [SPR16<sup>®</sup>] Springer Gabler | Springer Fachmedien Wiesbaden GmbH: *Gabler Wirtschaftslexikon*. <http://wirtschaftslexikon.gabler.de/Definition/maschinenbelegung.html> [Abruf: 2017-05-19]
- [VAN16<sup>®</sup>] W.M.P. van der Aalst: *Event Logs: What kind of data does process mining require?* <https://de.slideshare.net/wvdaalst/event-logs-what-kind-of-data-does-process-mining-require> [Stand: 2015-08-20] [Abruf: 2017-05-19]

## Veröffentlichungen des Verfassers

- [BHH+16<sup>#</sup>] J. Bernshausen, A. Haller, T. Holm, M. Hoernicke, M. Obst, **J. Ladiges**: *Namur Modul Type Package - Definition: Beschreibungsmittel für die Automation modularer Anlagen*. atp edition - Automatisierungstechnische Praxis, Vol. 58 (1-2), 2016, S. 72–81.
- [EWS+16<sup>#</sup>] T. Ernst, R. Weidner, K. Schwake, J. Wulfsberg, B. Hennings, R. Lammering, **J. Ladiges**, A. Fay: *Automatisierte, hochgenaue Fertigung und Montage an und in schwingenden Strukturen*. In: *Kongress Automation*, 2016.
- [FDT+15A<sup>#</sup>] A. Fay, C. Diedrich, M. Thron, A. Scholz, P. Puntel-Schmidt, **J. Ladiges**, T. Holm: *Wie bekommt Industrie 4.0 Bedeutung?* In: *Kongress Automation*, 2015.
- [FDT+15B<sup>#</sup>] A. Fay, C. Diedrich, M. Thron, A. Scholz, P. Puntel-Schmidt, **J. Ladiges**, T. Holm: *Wie bekommt Industrie 4.0 Bedeutung?* atp edition - Automatisierungstechnische Praxis, Vol. 57 (07-08), 2015, S. 30.
- [HCL+17<sup>#</sup>] C. Haubeck, A. Chakraborty, **J. Ladiges**, A. Pokahr, W. Lamersdorf, A. Fay: *Evolution of Cyber-Physical Production Systems supported by community-enabled experiences*. In: *IEEE 15th International Conference of Industrial Informatics (INDIN)*, 2017.
- [HHA+16<sup>#</sup>] T. Holm, U. Hempen, T. Albers, M. Obst, **J. Ladiges**, A. Fay: *DIMA: Method for the Integration of Modular Process Units*. In: *Annual Petroleum & Chemical Industry Committee Europe Conference (PCIC)*, 2016.
- [HLL+14A<sup>#</sup>] C. Haubeck, **J. Ladiges**, W. Lamersdorf, A. Fay: *Behandlung unbekannter Änderungen in automatisierten Produktionsprozessen anhand von Wissensmodellen*. In: *Software Engineering Workshops*, 2014.
- [HLL+14B<sup>#</sup>] C. Haubeck, W. Lamersdorf, **J. Ladiges**, A. Fay: *An active service-component architecture to enable self-awareness of evolving production systems*. In: *IEEE International Conference on Emerging Technology and Factory Automation (ETFA)*, 2014.
- [HLL+14C<sup>#</sup>] C. Haubeck, W. Lamersdorf, **J. Ladiges**, A. Fay, J. Fuchs, C. Legat, B. Vogel-Heuser: *Interaction of model-driven engineering and signal-based online monitoring of production systems: Towards Requirement-aware evolution*. In: *Conference of the IEEE Industrial Electronics Society (IECON)*, 2014.
- [HLW+16<sup>#</sup>] T. Holm, **J. Ladiges**, S. Wassilew, P. Altmann, A. Fay, L. Urbas, U. Hempen: *DIMA im realen Einsatz: Von der Idee zum Prototypen*. In: *Kongress Automation*, 2016.
- [HOL+16<sup>#</sup>] T. Holm, M. Obst, **J. Ladiges**, L. Urbas, A. Fay, T. Albers, U. Hempen: *Namur Model Type Package - Implementierung: Anwendung des Namur-MTP für Prozessanlagen*. atp edition - Automatisierungstechnische Praxis, Vol. 58 (1-2), 2016, S. 72–81.
- [HWB+13<sup>#</sup>] C. Haubeck, I. Wior, L. Braubach, A. Pokahr, **J. Ladiges**, A. Fay, W. Lamersdorf: *Keeping Pace with Changes - Towards Supporting Continuous Improvements and Extensive Updates in Production Automation Software*. *Electronic Communications of the EASST (Volume 56)*, 2013.
- [LFH+15<sup>#</sup>] **J. Ladiges**, A. Fülber, C. Haubeck, E. Arroyo, A. Fay, W. Lamersdorf: *Learning Material Flow Models for Manufacturing Plants from Data Traces*. In: *IEEE International Conference on Industrial Informatics*, 2015.
- [LFH+17<sup>#</sup>] **J. Ladiges**, A. Fay, T. Holm, U. Hempen, L. Urbas, M. Obst, T. Albers: *Integration of Modular Process Units into Process Control Systems*. In: *IEEE Transactions on Industry Applications, Vol. PP (99)* (angenommener Beitrag), 2017.
- [LFK+17<sup>#</sup>] **J. Ladiges**, A. Fay, A. Köcher, T. Holm, P. Altmann, L. Urbas: *Engineering und Überprüfung der gegenseitigen Abhängigkeiten von Diensten eines automatisierten Moduls*. In: *Kongress Automation*, 2017.

- [LFL16<sup>#</sup>] **J. Ladiges**, A. Fay, W. Lamersdorf: *Automated Determining of Manufacturing Properties and Their Evolutionary Changes from Event Traces*. *Intelligent Industrial Systems*, Vol. 2 (2), 2016, S. 163–178.
- [LHF+13<sup>#</sup>] **J. Ladiges**, C. Haubeck, A. Fay, W. Lamersdorf: *Operationalized Definitions of Non-Functional Requirements on Automated Production Facilities to Measure Evolution Effects with an Automation System*. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2013.
- [LHF+14A<sup>#</sup>] **J. Ladiges**, C. Haubeck, A. Fay, W. Lamersdorf: *Evolution Management of Production Facilities by Semi-Automated Requirement Verification*. at - *Automatisierungstechnik*, Vol. 62 (11), 2014, S. 781–793.
- [LHF+14B<sup>#</sup>] **J. Ladiges**, C. Haubeck, A. Fay, W. Lamersdorf: *Semi-automated decision making support for undocumented evolutionary changes*. In: *Workshop Software-Reengineering & Evolution (WSRE)*, 2014.
- [LHF+15<sup>#</sup>] **J. Ladiges**, C. Haubeck, A. Fay, W. Lamersdorf: *Learning Behaviour Models of Discrete Event Production Systems from Observing Input/Output Signals*. In: *IFAC/IEEE/IFIP/IFORS Symposium on Information Control Problems in Manufacturing (INCOM)*, 2015.
- [LHL+15<sup>#</sup>] **J. Ladiges**, C. Haubeck, S. Lity, A. Fay, W. Lamersdorf, I. Schäfer: *Supporting Commissioning of Production Plants by Model-Based Testing and Model Learning*. In: *International Symposium on Industrial Electronics (ISIE)*, 2015.
- [LHW+13<sup>#</sup>] **J. Ladiges**, C. Haubeck, I. Wior, E. Arroyo, A. Fay, W. Lamersdorf: *Evolution of Production Facilities and its Impact on Non-Functional Requirements*. In: *IEEE International Conference on Industrial Informatics (INDIN)*, 2013.
- [PBH+14<sup>#</sup>] A. Pokahr, L. Braubach, C. Haubeck, **J. Ladiges**: *Programming BDI Agents with Pure Java*. In: Müller, Weyrich, Bazzan (Hrsg.): *Multiagent System Technologies*: Springer International Publishing (Lecture Notes in Computer Science), Vol. 8732, S. 216–233, 2014.
- [SDD+<sup>#</sup>17] M. Schleipen, N. D'agostino, M. Damm, C. Ewertz, A. Gössling, R. Henßen, T. Holm, S. Hoppe, **J. Ladiges**, A. Lüder, O. Sauer, N. Schmidt, R. Wilmes: *Harmonisierung im Kontext Industrie 4.0 - AutomationML und OPC UA*. In: *Kongress Automation*, 2017.
- [VFF+15<sup>#</sup>] B. Vogel-Heuser, S. Feldmann, J. Folmer, M. Kowal, I. Schäfer, **J. Ladiges**, A. Fay, C. Haubeck, W. Lamersdorf, S. Lity, T. Kehrer, M. Tichy, S. Getir, M. Ulbrich, V. Klebanov, B. Beckert: *Selected Challenges of Software Evolution for Automated Production Systems*. In: *IEEE International Conference on Industrial Informatics*, 2015.
- [WLA+13<sup>#</sup>] I. Wior, **J. Ladiges**, E. Arroyo, A. Fay: *First steps from a traffic node to traffic networks — Modeling and stability*. In: *IEEE International Conference on Emerging Technologies & Factory Automation (ETFA)*, 2013.
- [WUL+16<sup>#</sup>] S. Wassilew, L. Urbas, **J. Ladiges**, A. Fay, T. Holm: *Transformation of the NAMUR MTP to OPC UA to allow Plug and Produce for Modular Process Automation*. In: *IEEE International Conference on Emerging Technology & Factory Automation (ETFA)*, 2016.

## Studentische Arbeiten

- [Bra15<sup>%</sup>] A. Brandenburg: *Untersuchung zur Anwendung von Methoden der künstlichen Intelligenz zur automatischen Generierung von Materialflussmodellen*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Masterthesis, 2015.
- [Fül13<sup>%</sup>] A. Fülber: *Bestimmung von Messprinzipien für Flexibilitätsanforderungen in Produktionssysteme*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Seminararbeit, 2013.
- [Fül14<sup>%</sup>] A. Fülber: *Anwendung und Evaluation von Mining-Techniken zur automatischen Generierung von Prozessmodellen für Fertigungsanlagen*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Masterthesis, 2014.