



DIPLOMARBEIT

Titel der Diplomarbeit

Simulationsmodell einer variablen Kleinserienfertigung
an Hand eines praktischen Beispiels

Verfasser

Martin Keiml

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften
(Mag. rer. soc. oec.)

Wien, im November 2008

Studienkennzahl lt. Studienblatt	157
Studienrichtung lt. Studienblatt	Internationale Betriebswirtschaft
Betreuer	O.Univ.Prof. Dipl.-Ing. Dr. Richard F. Hartl Univ.Ass. Dipl.-Ing. Dr. Christian Almeder

Erklärung der Selbstständigkeit

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.

Ich habe mich bemüht, sämtliche Inhaber der Bildrechte ausfindig zu machen und ihre Zustimmung zur Verwendung der Bilder in dieser Arbeit eingeholt. Sollte dennoch eine Urheberrechtsverletzung bekannt werden, ersuche ich um Meldung bei mir.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Stockerau, den 15.11.2008

Martin Keiml

Danksagung

Ich möchte mich bei all jenen bedanken, die direkt oder indirekt zur Entstehung dieser Diplomarbeit in Form fachlicher oder anderweitiger Unterstützung beigetragen haben.

Besonderer Dank gebührt dabei meinen Eltern, die mir durch ihre Unterstützung dieses Studium ermöglicht haben.

Weiters bedanke ich mich bei Herrn *Dipl. Ing. Dr. Christian Almeder* für die wissenschaftliche Betreuung meiner Arbeit. Obwohl ich sehr lange an dieser Arbeit schrieb war er stets sehr bemüht und eine große Unterstützung dabei, ein Simulationsmodell zu entwickeln.

Außerdem möchte ich mich bei Herrn *Ing. Josef Hahnl* sowie den Mitarbeitern der Firma M-U-T bedanken die mich fachlich unterstützen und mir die Möglichkeit gaben, neben meiner beruflichen Tätigkeit diese Arbeit fertigzustellen. Besonders danke ich Herrn *Günther Hanns*, der die Idee zu dieser Diplomarbeit gab und mich bei der Evaluierung und Auswertung der Daten unterstützte. Durch seine Hilfe konnten die Daten erst zu einem vernünftigen Ganzen strukturiert werden.

Mein Dank gebührt auch *Dr. Georg Sonneck* der mir bei der Umsetzung der Programmierarbeiten in Java half und mir die wesentlichen Konzepte der Programmierung näherbrachte.

Abkürzungen und Notationen

ABRA	Abfallbehälterreinigungsanlage
BDE	Betriebsdatenerfassung
CIM	Computer integrated manufacturing
CNC	Computer numeric control
CPM	Critical path method
CQN	Closed queuing network
FAZ	Früheste Anfangszeit
FES	Fertigungssteuerung
FEZ	Früheste Endzeit
FFS	Flexibles Fertigungssystem
FIFO	First in- first out
JIT	Just in time
KMU	Klein- und Mittelbetriebe
LIFO	Last in- first out
LP	Lineare Programmierung
MPS	Master production scheduling
MRP	Material resource planning
NC	Numeric control
OR	Operations research
PERT	Program evaluation and review technique
PPS	Produktionsplanungssystem
SAZ	Späteste Anfangszeit
SEZ	Späteste Endzeit
SKD	Semi-knocked down (tlw. vormontiert)

Inhaltsverzeichnis

Danksagung	IV
Abkürzungen und Notationen	VI
1 Einleitung	1
2 Aufgabenstellung	3
2.1 Problembeschreibung	3
2.2 Literaturhinweise	9
3 Gestaltung von Produktionssystemen	11
3.1 Strukturmodelle	14
3.1.1 Flexible Fertigungssysteme	15
3.1.2 Produktionsinsel	15
3.1.3 Einzelplatzmontage	17
3.1.4 Variable Fließfertigung	17
3.1.5 Kategorisierung der Montage bei M-U-T	18
3.2 Planungsmodelle	18
3.2.1 Johnsons Algorithmus	21
3.2.2 NEH- Heuristik	22
3.2.3 Heuristik für ein Mehrmaschinen- Problem	22
4 Methoden des Operations Research	25
4.1 Modellierungsschritte	27
4.2 Lineare Optimierung	31
4.3 Netzplantechnik	34
4.3.1 CPM- Netzpläne	35
4.3.2 PERT- Netzpläne	36
4.4 Warteschlangentheorie	38
4.4.1 Das Wartesystem M/M/1	41
4.4.2 Das geschlossene Warteschlangennetzwerk	43

4.4.3	Anwendungen für Warteschlangennetzwerke	43
4.5	Simulation	44
4.5.1	Anwendungsgebiete	44
4.5.2	Einteilung und Klassifizierung von Simulationsmodellen	45
4.5.3	Vor- und Nachteile der Simulation	48
4.5.4	Schritte der Simulationsmodellierung	51
4.6	Simulationssoftware	54
4.6.1	Simulations Sprachen	57
4.6.2	Simulationsanwendungen	57
5	Modellformulierung	59
5.1	Modellannahmen	60
5.2	Mathematische Grundlagen	61
5.2.1	Die Normalverteilung im Modell	62
5.2.2	Die Dreiecksverteilung im Modell	64
5.3	Prioritätsregeln	67
5.4	Datensammlung und Auswertung	68
5.5	Layouterstellung	70
5.6	Aktive Objekte	71
5.6.1	Objekt "Arbeitersource"	76
5.6.2	Objekt "Bearbeitung"	78
5.6.3	Objekt "Lackierarbeiten"	80
5.6.4	Objekt "Gemeinsame Weitergabe"	82
5.7	Entitäten	83
5.7.1	Kundenauftrags- Entitäten	84
5.7.2	Arbeitsauftrags- Entitäten	84
5.7.3	Arbeiter- Entitäten	85
5.8	Variablen	85
5.9	Weitere Steuerelemente	86
5.10	Verifizierung und Validierung	86
6	Untersuchungen	91
6.1	Design der Experimente	92
6.2	Durchführung und Auswertung der Experimente	93
6.3	Untersuchungsergebnisse und Diskussion	106
7	Zusammenfassung und Ausblick	108
	Literaturverzeichnis	112

A	Objekte des AnyLogic- Modells	113
A.1	Aktive Objekte	114
A.1.1	Objekt "Arbeitersource"	114
A.1.2	Objekt "Bearbeitung"	115
A.1.3	Objekt "Lackierarbeiten"	116
A.1.4	Objekt "Gemeinsame Weitergabe"	117
A.2	Entitäten	118
A.2.1	Kundenauftrags- Entitäten	118
A.2.2	Arbeitsauftrags- Entitäten	118
A.2.3	Arbeiter- Entitäten	118
B	Programmierung	125
B.1	Allgemeine Methoden für das Modell	125
B.2	Methode für das Objekt "Arbeitersource"	126
B.3	Methoden für das Objekt "Bearbeitung"	127
B.4	Methode für das Objekt "Lackierarbeiten"	131
C	Kurzfassung	134
D	Summary	136
E	Lebenslauf	138

Abbildungsverzeichnis

3.1	Ebenen für die Fertigungsplanung	19
3.2	Iterationsstufen beim CDS- Algorithmus	23
4.1	Graphische Lösung eines LP- Problems (vgl. [HL04], S.32) . .	32
4.2	Einfacher Netzplan	34
4.3	Arten von Netzplänen ([Dom07], S.99)	35
4.4	Einfacher CPM- Netzplan	37
4.5	Einfaches Warteschlangensystem	39
4.6	M/M/1- Warteschlangensystem ([HL04], S.781)	41
4.7	Klassifizierung von Simulationsmodellen ([VDI00])	47
4.8	Anlaufphase eines Simulationsmodells ([Prab], S.21)	52
4.9	Phasen einer Simulation	54
4.10	Ablauf eines Simulationsprozesses ([VDI00])	55
5.1	Dichtefunktion der Auftragsgenerierung	63
5.2	Dichteverteilung der Ausfallszeiten der Arbeiterentitäten	64
5.3	Dreiecksverteilung	65
5.4	Dreiecksverteilungen der Bearbeitungszeiten Rotopress an Station 7	66
5.5	Dreiecksverteilungen der Bearbeitungszeiten Variopress an Station 21	69
5.6	Der komplette Fertigungsprozess	72
5.7	Der komplette Fertigungsprozess (Forts.)	73
5.8	Der komplette Fertigungsprozess (Forts.)	74
5.9	Der komplette Fertigungsprozess (Forts.)	75
5.10	Ablaufdarstellung für das Objekt "Arbeiter"	77
5.11	Ablaufdarstellung für das Objekt "Bearbeitung"	79
5.12	Ablaufdarstellung für das Objekt "Lackierarbeiten"	81
5.13	Anteil der Ausfallszeit an der Gesamtarbeitszeit	89
5.14	Verteilung der Bearbeitungszeit beim Typ "Variopress"	90
6.1	Experiment 1: Auslastung der Arbeiter	94
6.2	Experiment 1: Ausfallszeiten der Arbeiter	95
6.3	Experiment 2: Auslastung der Arbeiter	98

6.4	Experiment 2: Ausfallszeiten der Arbeiter	99
6.5	Experiment 3: Auslastung der Arbeiter	101
6.6	Experiment 3: Ausfallszeiten der Arbeiter	102
6.7	Experiment 4: Auslastung der Arbeiter	104
6.8	Experiment 4: Ausfallszeiten der Arbeiter	105
A.1	Objekt Arbeitersource	115
A.2	Arbeiterobjekte im Modell	121
A.3	Statechart zur Steuerung Wahrscheinlichkeit der Ausfallszeiten	122
A.4	Objekt Bearbeitung	123
A.5	Objekt Lackieren	123
A.6	Objekt Gemeinsame Weitergabe	124

Tabellenverzeichnis

1.1	Bedeutung der Planung in Abhängigkeit vom Organisationstyp	2
2.1	Fahrzeugvarianten	6
2.2	Fertigungsstückzahlen der einzelnen Varianten	6
2.3	Montageschritte	8
4.1	Beispiele für Warteschlangensysteme	40
5.1	Lookup table "AuswahlFahrzeugtyp"	60
5.2	Lookup table "AuswahlVerkleidung" für Rotopress	60
6.1	Fertigungszeiten Experiment 1: Referenzmodell	93
6.2	Fertigungszeiten Experiment 2: Kapazitätsminimierung	97
6.3	Fertigungszeiten Experiment 3: Durchlaufzeitminimierung	101
6.4	Fertigungszeiten Experiment 4: Auftragsmaximierung	103
6.5	Vergleich der mittleren Durchlaufzeiten	106
A.1	Parameter des Modells	114
A.2	Parameter des Objekts Arbeitersource	116
A.3	Parameter des Objekts Bearbeitung	117
A.4	Parameter des Objekts Lackierarbeiten	118
A.5	Parameter der Entität Kundenauftrag	119
A.6	Parameter der Entität Arbeitsauftrag	120
A.7	Parameter der Entität Arbeiter	120

Kapitel 1

Einleitung

Das Ziel dieser Diplomarbeit ist es, ein Simulationsmodell zu entwickeln mit dem die -größtenteils manuelle- Endmontage in einem mittelständischen Maschinenbau- Unternehmen dargestellt werden kann. Dieses Simulationsmodell soll den komplexen Ablauf einer variablen Kleinserienfertigung abbilden, um damit Verbesserungspotentiale erkennen zu können. Eine dieser möglichen Verbesserungen ist dabei der Umstieg von Einzelplatzmontage auf eine Fertigungsstraße. Dies ist ein mögliches Szenario, welches im konkreten Fall untersucht werden soll. Die Verbesserung soll dabei vor allem hinsichtlich

- Minimierung der gesamten Durchlaufzeit und
- Maximierung der Kapazitätsauslastung

erkennbar sein. Das sind aber gegensätzliche Ziele. Eine Minimierung der Durchlaufzeit ist nur möglich, wenn die Kapazitäten entsprechend erweitert werden um Warte- und Liegezeiten zu vermeiden. Das geht zu Kosten der Kapazitätsauslastung die entsprechend sinkt.

Die Minimierung der Durchlaufzeiten und die Maximierung der Kapazitätsauslastung beschreiben danach gegenläufige Forderungen und somit konkurrierende Zielsetzungen. Parameter zur Lösung des Problems konkurrierender Ziele sind der Zeitpunkt, die Mengen, der Ort sowie die Kosten der eigentlichen Produktionsdurchführung ([Pet92], S.7).

Dieses Ziel optimal zu lösen ist in der kurzfristigen Planung nur durch den Einsatz rechnergestützter PPS- Systeme möglich. Die Termin- als auch Auftragssteuerung ist besonders planungsintensiv. Wie in Tab. 1.1 (vgl. [Pet92], S.13) ersichtlich ist die in diesem Fall vorliegende Form der Einzelfertigung mit teilweiser Serienfertigung in der Vormontage sehr planungsintensiv. Das

Strukturmerkmale Funktionsgruppe	Werkstatt- fertigung Einzel- fertigung	Reihen- fertigung Serien- fertigung	Fliess- fertigung Massen- fertigung
Mengenplanung	gering	groß	sehr groß
Terminplanung	sehr groß	groß	gering
Auftragssteuerung	sehr groß	groß	gering
Datenverwaltung	groß	sehr groß	gering

Tabelle 1.1: Bedeutung der Planung in Abhängigkeit vom Organisationstyp

gilt besonders bei kundenauftragsorientierter Fertigung da hier die Produkte nicht bevorratet werden können. Diese werden erst individuell auf Kundenwunsch gefertigt und müssen zum vereinbarten Liefertermin fertig sein. Hier müssen auch in der Fertigungsplanung Interdependenzen durch Einbeziehung aller Einflussgrößen vermieden werden.

Eine ähnliche Problemstellung behandelten Nyen, Bertrand, Ooijen und Vandaele ([NJON06]) für Lagerhaltungssysteme. Dieses Modell basiert ebenfalls auf Mehrprodukt- Mehrmaschinenfertigung mit stochastischen Ankunfts- und Bearbeitungszeiten. Für die strategische Planung bietet sich der Einsatz von Simulationssoftware an. Bei Simulation können auch gleichzeitig stattfindende Ereignisse und gegenseitige Beeinflussung von Parametern berücksichtigt werden.

Simultanplanungsmodelle[!] zeichnen sich dadurch aus, daß in ihnen sowohl die sachlichen Verflechtungen zwischen den unternehmerischen Funktionsbereichen als auch die zeitlich- dynamischen Verflechtungen der Handlungsabläufe abgebildet sind. ([Pet92], S.17)

Eine mögliche Layoutplanung für die Montage in der Produktionsstätte ist nicht vorgesehen. Dieses Simulationsmodell soll nur zur Darstellung und Optimierung des Montageflusses dienen.

Kapitel 2

Aufgabenstellung

2.1 Problembeschreibung

Als Praxisbeispiel dient die Fahrzeugmontage der Firma M-U-T in Stockerau. 1932 erfolgte die Gründung des Unternehmens 30 km nordöstlich von Wien mit dem Namen M-U-T als Kurzbezeichnung für Maschinen - Umwelttechnik - Transportanlagen. Zu Beginn beschäftigte man sich ausschließlich mit Förderanlagen. Anfangs wurde die Fertigung außer Haus gegeben, kurze Zeit später begann M-U-T mit einer eigenen Produktion. Heute steht eine Betriebsfläche von insgesamt 32.500 m², sowie eine Hallenfläche von 13.000 m² zur Verfügung. Der Förderanlagenbau ist auch heute noch eines der wichtigsten Standbeine der Fa. M-U-T.

1955 wurde mit dem Bau des ersten Müllwagens die Sparte Fahrzeugbau begründet. Heute ist M-U-T Anbieter einer umfangreichen Palette auf dem Gebiet der Kommunalfahrzeuge und kann sich durch sein langjähriges Know-how auch auf dem internationalen Markt erfolgreich behaupten. M-U-T ist mit seinem Produktionsprogramm der Sparte Fahrzeugbau Marktführer in Österreich und Ungarn.

Die Endmontage der unterschiedlichen Fahrzeugtypen erfolgt derzeit in einer einzigen Montagehalle in Einzelplatzmontage. Dieser Montageablauf soll nun in ein Simulationsmodell übersetzt werden um Schwächen und Verbesserungspotentiale im Ablauf erkennen zu können. Speziell eine bessere Auslastung, Verkürzung der Montagezeiten und damit auch eine kürzere Durchlaufzeit sind dabei vorrangige Ziele.

Auf Basis dieses Modells werden anschließend konkrete Experimente durch-

geführt um oben genannte Ziele erreichen zu können. Diese Experimente sollen auch Ereignisse wie Maschinenausfälle, Ausfälle von Personal (Urlaub, Krankenstand, Unfall,...) berücksichtigen.

Bei der Firma M-U-T werden unterschiedliche Kommunalfahrzeugaufbauten hergestellt und montiert. Die beiden Hauptprodukte sind die Abfallsammelaufbauten "Rotopress" sowie "Variopress".

Die Fertigung kann in drei Stufen unterschieden werden:

- Vorfertigung
- Hauptfertigung
- Endmontage

Die Endmontage erfolgt derzeit als Einzelplatzmontage. Dieses Modell wird nur auf die Endmontage eingehen und die Vor- und Hauptfertigung nicht berücksichtigen. Der Rotopress ist ein Abfallsammelfahrzeug mit rotierender Drehtrommel. Durch die Drehung wird der Müll in den Behälter gefördert und verdichtet. Dieses Abfallsammelfahrzeug findet vor allem in der Sammlung von Hausmüll und Biomüll Verwendung. Für die Sammlung von Biomüll gibt es eine eigene Ausführung aus rostfreiem Edelstahl da Biomüll sehr aggressiv ist und normalen Stahl rasch zerfressen würde. Die Montage sowie Montagezeiten für diese beiden Varianten sind identisch, allerdings müssen die entsprechenden Halbfabrikate verfügbar sein.

Weiters ist der Rotopress mit zwei Verkleidungsvarianten erhältlich. Der runden "Classic"- sowie der eckigen "Styling"- Verkleidung. Diese haben unterschiedliche Halbfabrikate sowie Montagezeiten.

Neben vielen verschiedenen Sonderzubehörvarianten ist speziell die Abfallbehälter- Reinigungsanlage ("ABRA") zu beachten. Damit ist es möglich, während der Abfallsammlung die Mülltonnen in einem Arbeitsgang zu entleeren und danach auch gleich auszuwaschen. Dadurch sind umfangreiche Einbauten notwendig. Neben den Wascharmern und Leitungen beispielsweise auch ein Reinwasser- und Schmutzwassertank. Diese Einbauten müssen sukzessive mit dem gesamten Aufbau durchgeführt werden.

Für den Rotopress stehen verschiedene zugekaufte Schüttungssysteme zur Verfügung. Diese können auch in Kombination montiert werden. Die Systeme werden im Werk zusammengebaut, lackiert und auf das Fahrzeug montiert.

Für die verschiedenen Schüttungen und Systeme ist der Montageaufwand sehr unterschiedlich.

Der Variopress ist ein Pressmüllwagen. Der Müll wird in das Fahrzeug geleert und von einem Pressschild gegen die Rückwand des Behälters gedrückt. Damit lässt sich eine wesentlich höhere Verdichtung erreichen. Der Variopress ist im Gegensatz zum Rotopress nicht flüssigkeitsdicht. Er findet vor allem in der Papier- und Sperrmüllsammlung Verwendung da auch Großbehälter entleert werden können.

Der Variopress ist auch in der Variante "Powerpress" erhältlich. Der Unterschied zwischen beiden Aufbauvarianten ist nur das Heckteil. Dieses ist beim Powerpress speziell verstärkt um vor allem bei der Sperrmüllsammlung genügend Kraft aufzubringen um den Sperrmüll zusammenzupressen. Deshalb sind auch die Montagezeiten für diese beiden Varianten unterschiedlich. Das Powerpress- Heckteil wird in einem anderen Werk gefertigt und die Heckteillagerung muss erst an den Behälter angepasst werden.

Für den Variopress stehen verschiedene Schüttungssysteme zur Verfügung. Diese können auch in Kombination montiert werden. Die Systeme werden im Werk zusammengebaut, lackiert und auf das Fahrzeug montiert. Für die verschiedenen Schüttungen und Systeme ist der Montageaufwand sehr unterschiedlich.

Weiters gibt es eine Vielzahl von Sonderzubehör das je nach Kundenwunsch eingebaut und montiert wird. Es gibt beim Variopress noch eine SKD- Ausführung (Semi- knocked down). Dabei wird nur der Behälter sowie das Heckteil im Werk montiert und das Fahrzeug grundiert. Sämtliche anderen Anbauteile werden anschließend in den Behälter geladen und das Fahrzeug zur Endmontage in ein anderes Werk überstellt. Die SKD- Variante hat im Vergleich zum Komplettaufbau ca. 170 Std. weniger Montagezeit.

Beide Fahrzeugtypen sind in jeweils vier verschiedenen Größen verfügbar. Diese vier Größen können hinsichtlich Montagezeiten vernachlässigt werden. Es sind also in der Simulation die in Tab. 2.1 genannten Varianten zu berücksichtigen die auch in Kombination auftreten können.

Die einzelnen Varianten kommen in unterschiedlichen Stückzahlen vor. Die Erfahrungen aus den letzten Jahren zeigen folgende Verteilungen:

Die Vorfertigung der einzelnen Bauteile erfolgt in Kleinserienfertigung

Rotopress	Variopress	Auswirkung
in Normal/ Bioausführung		Andere Halbfabrikate, Montagezeit ist gleich
mit Classic/ Styling-Verkleidung	Mit/ohne verstärktem Heckteil	Andere Halbfabrikate, andere Montagezeiten
mit/ohne Waschanlage		Zusätzlicher Montageaufwand während der gesamten Montage
	Komplettmontage oder SKD	Nur tlw. Montage im Werk
Kammschüttung	Balkenschüttung oder Kammschüttung	Unterschiedliche Schüttungen und Schüttungskombinationen sowie Montagezeiten

Tabelle 2.1: Fahrzeugvarianten

Geschäftsjahr	2004/05	2005/06	2006/07	2007/08
Rotopress "Classic"	0	7	38	14
Rotopress "Classic", Bioausf.	39	44	2	3
Rotopress "Classic", ABRA	0	7	0	0
Rotopress "Styling"	0	0	1	1
Rotopress "Styling", Bioausf.	23	23	18	22
Rotopress "Styling", ABRA	4	6	5	5
Variopress	33	32	20	36
Variopress, SKD	62	44	64	30
Powerpress	8	4	4	4
Powerpress, SKD	6	18	15	10
Summe	175	185	167	125
Andere Fahrzeugtypen	11	21	23	22
Gesamtproduktion	186	206	191	147

Tabelle 2.2: Fertigungsstückzahlen der einzelnen Varianten

mit 3 bis 10 Stück. Diese Halbfabrikate werden in einer Montagehalle auf das Chassis aufgesetzt und montiert. Jedes Fahrzeug belegt dabei einen Stellplatz für die komplette Dauer der Montage. Bei den Lackierarbeiten sowie Anpassungs- und Kontrollarbeiten bleibt der Stellplatz frei.

Die einzelnen Montageschritte sehen wie folgt aus:

	Rotopress	Variopress	Variopress SKD
1	Fahrgestell abschränken		
2	Hilfsrahmen anpassen		
3	Hilfsrahmen lackieren		
4	Hilfsrahmen, Hydraulik und Elektrik am FG montieren		Hilfsrahmen montieren
5	Behälter auf Bock zusammenbauen (parallel zu Hilfsrahmen anpassen u. lackieren), alle Teile bereits lackiert, Hydraulik am Aufbau montieren	Hydraulik am Behälter montieren	
6	Deckel an Behälter anpassen, Konus einschweißen, Deckel abhängen		
7	Behälter auf Fahrgestell aufsetzen		
8	Deckel lackieren (parallel zu Behälter aufsetzen)	Ausstoßschild in Behälter montieren	
9			Teile in Behälter einladen
10	Elektrik am Deckel verlegen	Heckteil und div. Einbauten zusammenbauen, Hydraulik und Elektrik verlegen (parallel zu Ausstoßschild montieren)	Heckteil und div. Einbauten zusammenbauen, Hydraulik verlegen (parallel zu Ausstoßschild montieren)
11	Deckel auf Behälter montieren	Heckteil auf Behälter montieren	

12	Elektrik und Hydraulik zusammenschließen		Verkabelung der Heckleuchten, Hydraulik zusammenschließen
13	Kipper auf Lackiergestell aufbauen (parallel zu Deckelmontage)	Kipper auf Lackiergestell aufbauen (parallel zu Heckteilmontage)	
14	Kipper lackieren (parallel zu Deckelmontage)	Kipper lackieren (parallel zu Heckteilmontage)	
15	Kipper montieren		
16	Verkleidung montieren		
17			Hydrauliköl füllen
18	Funktionskontrolle		
19		Fahrzeug lackieren	
20	Endkontrolle		
21	Nacharbeiten		
22	Fahrzeug reinigen und wiegen		Fahrzeug wiegen
23	Auslieferung an Kunden		Auslieferung f. Endmontage

Tabelle 2.3: Montageschritte

Diese Montageschritte sollen nun strukturiert und auf einzelne Stationen aufgeteilt werden. Ziel ist es, ein Simulationsmodell zu erstellen das alle verschiedenen Fertigungsvarianten berücksichtigt und mit dem die Durchlauf- und Fertigungszeit optimiert wird.

Das ist wegen der Variantenvielfalt in der Produktpalette sowie den verschiedenen Montagezeiten in den Stationen schwierig. Der gleiche Arbeitsschritt kann bei verschiedenen Modellen sehr unterschiedliche Montagezeiten aufweisen. Weiters fallen einige Arbeitsschritte nur bei bestimmten Varianten an wodurch es möglich sein muss, innerhalb der Fertigung andere Fertigungsaufträge zu überholen wenn man diese als (getaktete) Fertigungsstraße umsetzen will.

Derzeit wird in Einzelplatzmontage gefertigt. Der Hilfsrahmen wird auch jetzt schon teilweise in einer anderen Halle angepasst und anschließend zum

Lackieren gebracht.

Auf dem Montageplatz wird der lackierte Hilfsrahmen auf das Fahrgestell aufgesetzt. Auf einem weiteren Montageplatz wird währenddessen schon der Behälteraufbau auf Bock vormontiert. Der vormontierte Behälter wird auf das Fahrgestell aufgesetzt. Gleichzeitig mit dem Aufsetzen wird beim Rotopress schon der Deckel lackiert. Der Deckel bzw. das Heckteil werden auf das Fahrzeug montiert. Parallel dazu wird bereits die Schüttung auf einem Lackiergestell vormontiert und lackiert. Nachdem Hydraulik und Elektrik miteinander verbunden sind wird die Schüttung montiert und die Funktions- und Endkontrollen durchgeführt. Vor der Auslieferung wird das Fahrzeug zum Schluss noch auf einer externen Brückenwaage verwogen.

Der Rotopress wird bereits aus komplett fertig lackierten Teilen montiert, der Variopress wird aus grundierten Teilen zusammengebaut und erst nach dem kompletten Zusammenbau und der Verkabelung lackiert.

Bei der SKD- Version wird nur die Hydraulik verlegt und das Heckteil montiert. Alle anderen Teile werden in das Fahrzeug verladen und das Fahrzeug in grundiertem Zustand in ein anderes Werk zur Endmontage überstellt.

2.2 Literaturhinweise

Mit der Produktivitätsanalyse in der Einzel- und Kleinserienfertigung hat sich auch Ocker ([Ock85]) auseinandergesetzt . Dabei bietet er eine genaue Einteilung und Übersicht über Produktivitäts- und Flexibilitätsfaktoren. Er stellt wesentliche Einflussgrößen dar und analysiert auch die Auswirkungen an Hand praktischer Beispiele. Dabei geht er aber immer nur von einfachen Stückgütern aus (Dreh- und Frästeilen) und zeigt die Auswirkungen von Maschinenbelegung und den Einsatz NC- gesteuerter Maschinen. Der manuelle Montageprozess wird dabei aber nicht analysiert und näher betrachtet.

Einen theoretischen Überblick zur Problematik der Produktionsplanung der Einzel- und Kleinserienproduktion bietet Aspion ([Asp00]). Er beschäftigt sich primär mit den Aufgaben der kurzfristigen operativen Produktionsprogrammplanung. Von Interesse ist dabei, dass er dabei im Gegensatz zu den meisten Lehrbüchern keine teileorientierte sondern eine auftragsorientierte Sicht wählt. Er geht auch auf die Koordinationsproblematik innerhalb der Lieferkette ein. Dabei beschränkt er sich nicht auf klassische PPS- Systeme sondern versucht auch mathematische Lösungen mittels Warteschlangentheo-

rie und Clusteranalyse zu erarbeiten. Weiters bietet er einen Überblick über durchgeführte Simulationsstudien sowie deren Ergebnisse.

Ein ähnliches wie das hier formulierte Problem hat Tiedt in seiner Arbeit "Ein Beitrag zur Effektivitätssteigerung flexibler Fertigungssysteme durch Simulation und Optimierung der Einlastungsreihenfolge" behandelt. [Tie92]

Er löst dabei das Problem einer FFS- Leiterplattenfertigung mit dem Simulationsprogramm ROSI. Mittels einer Gruppenregel wird die zu simulierende Einlastungsreihenfolge minimiert und dadurch kann die optimale Einlastungsfolge ermittelt werden.

Im vorliegenden Fall ist eine Optimierung über die Einlastungsreihenfolge aber nicht möglich, da die Aufträge zumeist einzeln und in größeren Abständen eingehen. Auf Grund der langen Bearbeitungszeiten sowie der Variantenvielfalt kann auch nicht gewartet werden, bis eine bestimmte Stückzahl einer Variante erreicht ist bevor diese in die Fertigung geht. Das würde für den Kunden eine Verzögerung der Lieferung von mehreren Wochen bedeuten. Es kann auch kein deterministisches FFS- Modell angenommen werden da es sich hier um ein stochastisches Problem mit Fertigungsinseln handelt.

Weiters setzt die Arbeit von Tiedt auf der PPS- Ebene an während sich die vorliegende Arbeit auf der strategisch- taktischen Planungsebene bewegt.

Scholl [Sch95] versucht exakte Lösungen mittels Branch and Bound sowie LP- Modellen für die Optimierungen von Montagebändern zu bieten. Er geht dabei sowohl auf die Einprodukt- als auch Mehrproduktfließfertigung ein. Die Mehrproduktfließfertigung lässt sich auch für diese Aufgabenstellung anwenden. Er behandelt beispielhaft kleinere Problemstellungen die er auch mittels Software exakt lösen kann. Damit kann nicht die Komplexität dieser Aufgabenstellung erfasst werden. Auf Grund der Größe dieses Modells wäre es für LP- Modelle np- hard und damit nicht lösbar.

Es gibt auch eine Fülle von Arbeiten, die sich mit der automatisierten Fertigung sowie der Auslegung und Simulation solcher Anlagen befassen. Die flexiblen Fertigungsanlagen in diesen Arbeiten sind jedoch durchwegs mit Robotern bestückt und beschäftigen sich mit der Serienproduktion (siehe [War96]). Auch das greift nicht das Problem der manuellen Kleinserienproduktion auf.

Kapitel 3

Gestaltung von Produktionssystemen

In der heutigen Zeit gibt es eine Vielzahl von verschiedenen Montage- und Fertigungsmodellen. Die Bandbreite reicht dabei von der klassischen Einzelplatz- oder Werkstättenfertigung bis hin zur vollautomatisierten Fertigungsstraße mit Robotern. Das gewählte Fertigungsmodell spiegelt sich dabei in der Regel auch in der Firmengröße wieder.

Während Kleinst- und Kleinbetriebe oft sehr spezialisierte Aufträge in Handarbeit erledigen bedienen Konzerne mit Massenartikeln den Markt für Standardprodukte. Dazwischen stehen die KMU's die sich für eine passende Methode entscheiden müssen.

Kleinere KMU's werden auf Grund ihres spezialisierten Produktportfolios dabei zur Einzel- oder Kleinserienfertigung tendieren. Ab einem gewissen Produktionsvolumen macht es Sinn, in Richtung standardisierter Fertigungsprozesse zu gehen. Gleichzeitig soll man dabei flexibel genug bleiben, um individuelle Kundenwünsche umsetzen zu können. Hier bietet sich an flexible Fertigungssystemen (FFS) bzw. Fertigungsinseln einzusetzen.

Fertigungsinseln können als "Fabrik in der Fabrik" angesehen werden. In ihnen erzeugen Mitarbeitergruppen ... bestimmte hinsichtlich ihres Ressourcenbedarfs und ihrer Fertigungsprozesse ähnliche Produktarten. Flexible Fertigungssysteme sind Systeme, in denen man Fertigungsstationen zusammenfaßt, die Produkte mit ähnlichen Fertigungsprozessen bearbeiten. Konkret handelt es sich um Systeme numerisch gesteuerter Werkzeugmaschinen (sog. NC-Maschinen), die durch Werkzeugwechsel sehr rasch umgerüstet

werden können und durch ein automatisches Transportsystem miteinander verbunden sind. ([Sch02], S.13)

Entsprechend handelt es sich hier um Fertigungsinseln da die Endmontage größtenteils händisch und ohne den Einsatz von NC- Maschinen erfolgt.

Die Einteilung und Klassifizierung in Produktionstypen ist hilfreich für ein strukturiertes Herangehen an eine gesetzte Aufgabenstellung.

Der Zweck einer Bildung von Produktionstypen liegt einmal in der Beschreibung der Realität. Jeder in der Wirklichkeit vorkommende Produktionsprozess kann anhand seiner Merkmale einem bestimmten Produktionstyp zugeordnet werden. Hierbei versucht man, die für den jeweiligen Untersuchungszweck maßgeblichen Merkmale zu verwenden. [EI93a], S.1718

Die Erscheinungsform des vorliegenden Betriebes kann nach [GT05] vorgenommen werden. Zu den **Produkt- und Programmeigenschaften** zählen:

- Güterart: mehrteilige, bewegliche Stückgüter
- Mehrproduktproduktion
- Einzel- und Serienproduktion
- Kundenproduktion

Da in diesem Praxisbeispiel verschiedene Varianten von Fahrzeugen gefertigt werden handelt es sich eindeutig um mehrteilige, bewegliche Stückgüter. "Beweglich" bezieht sich in diesem Fall jedoch nicht auf die Fahreigenschaften des Fahrzeuges selbst, sondern besagt, dass das Produkt selbst manipuliert werden kann. Bewegliche Güter werden an einem Produktionsstandort gefertigt und dann zum Bestimmungsort gebracht. Unbewegliche Güter wären beispielsweise Gebäude, Förderanlagen oder Fabriken. Diese werden vor Ort in Baustellenmontage errichtet. Wie in Tab. 2.1 ersichtlich werden auch mehrere Produkte gefertigt. Dies erfolgt in Kleinserienproduktion direkt für den Kunden. Ohne einen konkreten Auftrag eines Kunden wird also kein Fahrzeug gefertigt. Im Gegensatz dazu steht bei der Massen- oder Serienfertigung die Produktion für den anonymen Markt.

Die Einteilung kann nicht nur aus Sicht des Produktprogrammes sondern auch aus Sicht der Fertigungsorganisation erfolgen. Bezogenen auf den **Fertigungsprozess** erfolgt die Vor- und Hauptfertigung funktional in Werkstattproduktion. Gleichartige Arbeitsgänge werden dabei räumlich in einer

Werkstätte/ Produktionshalle zusammengefasst. Es gilt also das Funktionsprinzip.

In der Werkstattproduktion können Arbeits- und Transportvorgänge nicht vollkommen abgestimmt und optimiert werden. Auf Grund von wechselnden Arbeitsaufträgen für die verschiedenen Folgeaufträge, verschiedenen Losgrößen usw. kommt es zu Zwischenlagerständen. In der Folge entstehen Transportprobleme beim innerbetrieblichen Weitertransport da dieser normalerweise manuell mit Transportwagen und Staplern und nicht über automatische Transportanlagen wie Förderbänder erfolgt. Damit entstehen auch Leerzeiten bei einzelnen Maschinen während andere überbelegt sind.

Bei Anwendung des Objektprinzips orientiert sich die Anordnung der Arbeitssysteme an den Arbeitsplänen der zu bearbeitenden Erzeugnisse. Es können verschiedene Organisationstypen der Produktion unterschieden werden, die auf dem Objektprinzip basieren. [GT05], S.15

In der Endmontage herrscht das Objektprinzip vor. Hier kann zwischen Reihenproduktion und Zentrenproduktion in Form von Produktionsinseln unterschieden werden. Bei der Reihenproduktion ist der Ablauf für alle Produkte weitgehend identisch und es erfolgt keine zeitliche Bindung. Es handelt sich dabei meist um die Massengütererzeugung bei der auch hohe Lager- und Zwischenlagerbestände aufgebaut werden. Dieses System wird in Kapitel 3.1.4 behandelt während auf die Produktionsinseln in 3.1.2 eingegangen wird.

Der **Produktionsprozess** selbst lässt sich beschreiben als:

- Konvergierender Materialfluss: Ein Produkt wird aus mehreren Werkstoffen/ Produktgruppen hergestellt.
- Diskontinuierlicher Materialfluss: Der Zeitpunkt der Weitergabe ist nicht bei allen Arbeitsstationen gleich.
- Mehrstufige Produktion: Mehrere Arbeitsgänge sind notwendig wobei einzelne Arbeitsgänge in Stufen zusammengefasst werden können. Diese müssen abgeschlossen sein bevor mit den nächsten Arbeitsgängen begonnen werden kann.
- Veränderbare Arbeitsgangfolge: Einzelne Arbeitsgänge können umgruppiert werden.

- Arbeitsintensive Produktion: Die Arbeitsleistung hat im Vergleich zum Materialeinsatz einen sehr hohen Anteil.

Bezogen auf den **Fertigungsablauf** kann entsprechend obiger Einteilung auch zwischen der Flow-Shop- und der Job-Shop Fertigung unterschieden werden (vgl. [Sch02], S.14ff):

- Bei der Flow-Shop-Fertigung durchlaufen alle Fertigungsaufträge dieselbe Folge von Bearbeitungsstationen. Diese Stationen können in der Werkstatt-, Verrichtungsgruppen- oder auch Fließanordnung vorliegen. Häufig wird hierbei der Materialfluss durch Pufferläger geleitet, die bisweilen sogar die Möglichkeit eröffnen, die Reihenfolge der Aufträge zu verändern.
- Bei der Job-Shop-Fertigung ist die Reihenfolge der zu durchlaufenden Bearbeitungsstationen nicht notwendigerweise identisch. Jeder Auftrag sucht sich seine "eigene Bahn". Bei der Job-Shop- Fertigung ist die Reihenfolge der Aufträge in hohem Maße variabel. Der Job-Shop- Ablauf kann als eine Überlagerung von Flow-Shop- Abläufen angesehen werden.

Die Fertigung bei M-U-T kann als Mischform angesehen werden. Prinzipiell ist der Ablauf der Fertigung wie beim Flow-Shop vorgegeben. Es gibt allerdings nur parallele Arbeitstationen/ Montageplätze an denen alle Bearbeitungsschritte für ein Fahrzeug ausgeführt werden. Es gibt also nur eine Bearbeitungsstation die sich nur mit den einzelnen Lackiervorgängen abwechselt. Das Pufferlager ist ein gemeinsamer Parkplatz vor der Halle wo die angelieferten Fahrgestelle und auch fertigen Fahrzeuge parken. Wenn ein Fahrzeug nicht weiter bearbeitet werden kann weil Teile fehlen, dann kann auch das halbfertige Fahrzeug zwischengeparkt werden und es wird an einem anderen Fahrzeug weitergearbeitet. Oder es werden, soweit möglich, andere Arbeitsschritte vorgezogen. Diese individuelle und flexible Vorgehensweise bei Ablaufproblemen würde der Job-Shop-Fertigung entsprechen.

3.1 Strukturmodelle

In einer Simulationsstudie haben Drolet, Marcoux und Abdalnour [DMA08] einen Vergleich zwischen flexiblen Fertigungssystemen, Produktionsinseln und der klassischen Einzelplatzmontage angestellt. Diese Studie zeigt, dass Fertigungszellen der klassischen Werkstattfertigung hinsichtlich durchschnittlichem und maximalem Durchfluss, Auslastung und Verspätung überlegen sind. Im Folgenden werden die verschiedenen Fertigungssysteme vorgestellt und die jeweiligen Vor- und Nachteile dargestellt.

3.1.1 Flexible Fertigungssysteme

Ein flexibles Fertigungssystem (FFS) ist ein "Produktionstyp, der als höchste Stufe der Automatisierung in der Serienfertigung aus einem System verketteter Einzelmaschinen besteht, die in einer nicht durch Rüstvorgänge unterbrochenen Folge von Bearbeitungsvorgängen verschiedene (fertigungsverwandte) Werkstücke gleichzeitig bearbeiten können. (...) Im Gegensatz zu Transferstraßen, bei denen eine starre Verkettung mehrerer Einzelmaschinen und damit ein bestimmter Materialfluss realisiert ist, lassen sich flexible Fertigungssysteme dadurch kennzeichnen, dass eine EDV-Anlage die Steuerung der Bearbeitungsmaschinen und des Materialflusssysteme übernimmt. [EI93b], S.701

Flexible Fertigungssysteme setzen einen hohen Automatisierungsgrad voraus, insbesondere hinsichtlich des Materialflusses. Ein FFS besteht normalerweise aus mehreren NC- gesteuerten Maschinen die durch ein gemeinsames Transportsystem beschickt werden. Der Grad der Automatisierung kann dabei von manueller bis zu vollautomatischer CNC- Bedienung mittels Roboter und Programmierung durch CIM- Systeme reichen.

Normalerweise sind alle Vorgänge in einem FFS- System durch einen zentralen FFS- Zellenrechner gesteuert. Dieser ist oft auch noch in eine Rechnerhierarchie eingebettet und mit anderen FFS- Zellenrechnern oder dem PPS- System verbunden.

Durch diesen Rechner ist es möglich, die Rüstzeiten innerhalb des FFS zu minimieren. In einem FFS sind in der Regel eine bestimmte Anzahl voreingestellter Werkzeuge und Werkzeugkonfigurationen definiert. Die Werkzeuge befinden sich in einem zentralen Werkzeuglager und/ oder in lokalen Werkzeugmagazinen direkt an der Maschine. Kombiniert mit automatisierten Spannplätzen (die gleichzeitig auch Systemgrenze sein können) ist damit ein rascher Wechsel zwischen verschiedenen Fertigungsaufträgen möglich.

FFS sind bei der Serienproduktion zu finden. Umrüstvorgänge erfolgen auf Grund der Automatisierung rasch und die verschiedenen NC- Programme sind sofort verfügbar. Für individuelle Fertigung in Kleinstserien oder Einzelfertigung sind sie jedoch ungeeignet.

3.1.2 Produktionsinsel

Wird auf vollständige Automatisierung sowie den Einsatz von NC- Maschinen verzichtet, dann spricht man von einer Produktionsinsel. Produktionsinseln sind teilautonome Arbeitsgruppen innerhalb eines größeren Gefüges wie

z.B. einer Werkstätte oder Produktionshalle. Sie werden als wesentlicher Bestandteil einer schlanken Produktion (lean production) angesehen. Produktionsinseln bestehen zumeist aus spezialisierten Arbeitern, die zugewiesene Produktionsaufträge im Team ausführen. Auf Grund dieser Organisation ist es möglich, mit geringem Planungs- und Koordinationsaufwand durch eine zentrale Steuerung rasch und effizient zu agieren.

Als Vorteil wird i.a. der größere Überblick über den aktuellen Produktionsfortschritt bei den Mitarbeitern einer Produktionsinsel im Vergleich zur zentralen Produktionsplanung und -steuerung angesehen. So kann eine überlappte Produktion von Aufträgen (offene Produktweitergabe) i. d. R. problemlos realisiert werden. Dadurch lassen sich beträchtliche Reduzierungen der Durchlaufzeit erreichen. Bei Störungen können die Mitglieder der Arbeitsgruppe eigenverantwortlich über die Maßnahmen zu ihrer Beseitigung entscheiden. Kommt es zu einer kurzfristigen Überlastung eines Arbeitsplatzes, dann kommen unterausgelastete Kollegen kurzfristig zu Hilfe. [GT05], S.246

Produktionsinseln sind auch in der vorliegenden Firma zu finden. In der Endmontagehalle sind Endmonteure verschiedener Profession (Schlosser, Elektriker, Schweißer) zu finden die jeweils auf verschiedene Fahrzeugtypen spezialisiert sind. Eine exakte Anleitung durch die FES oder die Hallenmeister ist dabei nicht mehr notwendig.

Jedes Team hat einen mehr oder weniger genau abgegrenzten Bereich mit einigen Stellplätzen. Auf diesen werden dann autonom die vorgesehen Fahrzeuge montiert. Jedes Team ist dabei auf einige wenige Varianten/ eine Erzeugnisfamilie spezialisiert. Die einzelnen Stellplätze können deshalb auch Lager- bzw. Montageplätze der Produktionsinsel interpretiert werden.

Damit ist auch die Definition der Werkstattfließfertigung erfüllt:

Alle Produkte durchlaufen die Werkstätten in derselben Reihenfolge. Jedoch können sie innerhalb der Werkstätten von unterschiedlichen Arbeitsträgern bearbeitet werden. Die Anzahl möglicher Transportbeziehungen zwischen den Arbeitsträgern ist daher größer als die bei Fließfertigung. [EI93a], S. 2346.

Alle Halbfabrikate und Teile laufen zum Schluss in der Endmontage zusammen. Innerhalb der Endmontagehalle werden die unterschiedlichen Endprodukte und deren Varianten nun von den Arbeitern montiert. Dabei kann die

Reihenfolge der Bearbeitungsschritte sowie die Bearbeitungsfolge der Aufträge selbst variieren. Damit herrscht wieder die Flexibilität der Produktionsinsel.

3.1.3 Einzelplatzmontage

Die Endmontage kann aber auch als Einzelplatzmontage angesehen werden. Im Rahmen von Einzelplatzmontage montieren dann Teams aus zwei bis drei Monteuren gemeinsam das Fahrzeug. Das Fahrzeug bleibt an seinem vorgesehenen Stellplatz stehen während die benötigten Arbeiter sowie Teile zum Stellplatz gebracht und montiert werden. Wenn auf Grund von Materialmangel oder anderer Montageschritte am Fahrzeug nicht weitergearbeitet werden kann, dann geht das Team zum nächsten Stellplatz weiter und führt dort die nächsten Montageschritte durch.

Da im Prinzip nur Handwerkzeuge verwendet werden sowie die Stellplätze unmittelbar nebeneinander angeordnet sind ist diese Betrachtungsweise ebenfalls möglich. Für die Einzelplatzmontage spricht ebenfalls, dass die Montagetams das gesamte Fahrzeug vom ersten bis zum letzten Schritt montieren. Auch das entspricht der klassischen Einzelplatzmontage.

3.1.4 Variable Fließfertigung

Um die Endmontage in Zukunft effizienter durchführen und damit die Durchlaufzeiten zu minimieren soll die Möglichkeit einer variablen Fließfertigung evaluiert werden. Die Fließfertigung ist ein Produktionstyp,

bei dem die Arbeitsträger dem Fließprinzip nach der Reihenfolge der Arbeitsgänge in Fertigungslinien angeordnet sind. Nach dem Grad der Leistungsabstimmung zwischen den verschiedenen Arbeitsträgern (Stationen) einer Fertigungslinie kann man zwischen Fließreihen, Fließstraßen, Fließbändern und Transferstraßen unterscheiden.

Bei Fließreihen- oder Reihenfertigung sind die Stationen einer Fertigungslinie nur lose miteinander verkettet. Sie müssen nicht mit genau derselben Geschwindigkeit arbeiten. Zwischen den Stationen stehen deshalb Pufferlager. [EI93b], S.702

Der Umstand der Reihenfertigung liegt auch hier vor. Die derzeitige Fertigung nach dem Muster der Produktionsinsel (siehe 3.1.2) soll entsprechend

der einzelnen Arbeitsfolgen aus Tabelle 2.1 in einzelne Stationen einer Fließfertigung aufgeteilt und miteinander verkettet werden. Auf Grund der unterschiedlichen Dauer der einzelnen Arbeitsschritte werden zwischen den Stationen Pufferlager eingerichtet.

3.1.5 Kategorisierung der Montage bei M-U-T

Die Montage der Fahrzeuge ist sehr komplex und flexibel. Der Automatisierungsgrad ist sehr niedrig, beinahe die gesamte Montage erfolgt von Hand. Damit sind CIM- Systeme nicht anwendbar. Anstelle der in der Literatur hinlanglich behandelten FFS- Systeme liegt ein System aus Produktionsinseln vor. Die Montage bei M-U-T lasst sich also zusammenfassend klassifizieren:

- auftragsbezogene Kleinserienfertigung
- manuelle Fertigung
- Objektprinzip ohne zeitliche Bindung
- Reihenfertigung mit Pufferlagern

Hierfur muss nun ein geeigneter Algorithmus gefunden werden um die Aufgabenstellung zu losen.

3.2 Planungsmodelle

Bei der Auswahl eines geeigneten Planungsmodells ist es erforderlich, die entsprechende Planungsebene sowie die Parameter festzulegen. Planungsmodelle existieren von der strategischen uber die taktische bis zur operativen Ebene. Planungsmodelle sollen durch eine geeignete Einlastungsreihenfolge und Planung der Fertigung diese optimieren und damit die Produktivitat verbessern.

Planungsmodelle sind wie in Abb. 3.1 ersichtlich oft hierarchisch aufgebaut. Zu den Planungsmodellen der strategischen Ebene gehort die Schatzung zukunftiger Bedarfe fur einen festgelegten Zeitraum. Aus diesen geschatzten Verkaufszahlen lasst sich eine aggregierte Produktion sowie der Personalbedarf grob abschatzen. Daraus wird dann ein Produktionsplan (MPS, master production schedule) entwickelt. Dieser stellt die Verbindung zur taktischen Planungsebene her und wird mit der Materialplanung (MRP, master resource planning) verfeinert.

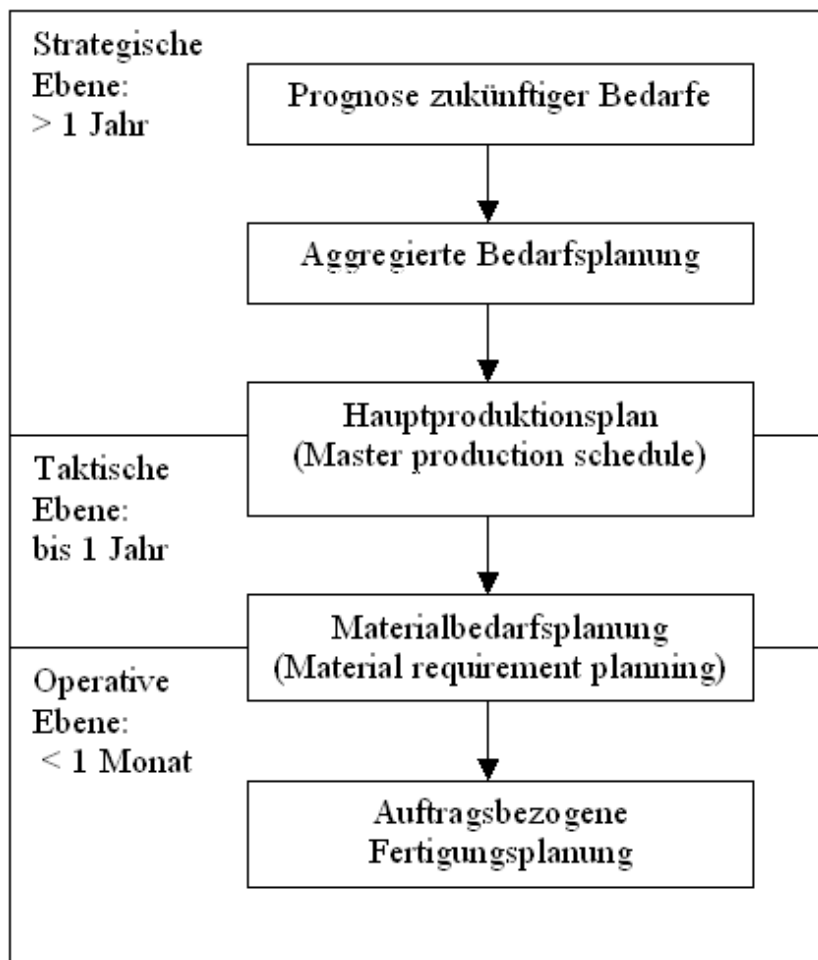


Abbildung 3.1: Ebenen für die Fertigungsplanung

Die aktuellen Fertigungsaufträge werden dann letztendlich auf der operativen Planungsebene eingelastet. Hier existiert eine Vielzahl verschiedener Modelle für die verschiedensten Anwendungsfälle der Produktion.

Neben den exakten Methoden sowie Heuristiken die für die zahlreichen Modelle existieren besteht auch noch die Alternative der Simulation. Diese kann auf allen Ebenen der Planung eingesetzt werden. Da Simulationsmodelle in der klassischen Form jedoch niemals eine exakte Lösung bieten können sollte immer versucht werden, ein exaktes Modell für die Optimierung zu entwickeln und einzusetzen.

Die klassische MRP- Planung ist für diese Aufgabenstellung nicht ausreichend und zu allgemein gehalten. Hier geht es um die effektive Bewertung und

Verbesserung eines bestehenden Fertigungssystemes. Für eine solche Analyse ist ein Planungsmodell der taktischen Ebene zu grob. Deshalb muss auf ein operatives Planungs- oder Simulationsmodell zurückgegriffen werden.

Für die Aufgabenstellung dieser Arbeit muss ein Modell die folgenden Kriterien erfüllen:

- Mehrere unterschiedliche Produkte werden gefertigt
- Der Fertigungsplan ist nicht exakt bekannt
- Mehrere verschiedene Arbeitsplätze sind notwendig
- Mehrere verschiedene Ressourcen werden eingesetzt
- Die Durchlaufzeit soll minimiert werden.

Für eine optimale Lösung muss ein exaktes Verfahren verwendet werden. Dazu bietet sich lineare Optimierung (siehe Kapitel 4.2) oder nicht lineare Optimierung an. Auch andere Methoden des Operations Research können verwendet werden. Für die exakte Lösung müssen die Daten allerdings in deterministischer Form vorliegen. Da in diesem Modell auch stochastische Einflussgrößen berücksichtigt werden müssen ist eine exakte Lösung nicht möglich. Methoden des Operations Research werden trotzdem in Kapitel 4 vorgestellt. Die Modellierung und Modellerstellung ist im wesentlichen, unabhängig von der jeweiligen Lösungsmethode, gleich und kann deshalb auch für diese Arbeit angewendet werden. Weiters werden Unterschiede und Anwendungsmöglichkeiten ausgewählter Methoden herausgearbeitet und mit den Anforderungen dieses Modells verglichen.

Werden die stochastischen Anforderungen dieses Modells vernachlässigt und durch deterministische Werte ersetzt so würden sich neben der exakten Lösung mit OR- Methoden auch Heuristiken anbieten. Diese liefern keine exakte Lösung, sind allerdings wesentlich einfacher zu erstellen und liefern rasch und mit geringem Rechenaufwand Lösungen die in der Nähe des Optimums liegen. Einer dieser Heuristiken die unter diesen Annahmen in Frage kommen ist die Mehr- Maschinen Heuristik von Cambell, Dudek und Smith die diese 1970 entwickelt haben. Diese Heuristik optimiert die Einlastungsreihenfolge mehrerer Aufträge auf m hintereinanderfolgenden Maschinen mit dem Ziel, die Durchlaufzeit zu minimieren. Sie basiert auf dem Johnson- Algorithmus der die Einlastungsreihenfolge von Aufträgen für ein Zwei- Maschinenproblem löst.

3.2.1 Johnsons Algorithmus

Der Johnson- Algorithmus betrachtet eine gegebene Anzahl von Aufträgen, die nacheinander auf zwei Maschinen bearbeitet werden müssen. Alle Aufträge müssen die Maschinen in der selben Reihenfolge passieren und die Bearbeitung auf Maschine 2 kann erst beginnen, wenn die Bearbeitung auf Maschine 1 abgeschlossen ist. Der Johnson- Algorithmus hat zum Ziel die Durchlaufzeit aller Aufträge zu minimieren. Damit wird auch die durchschnittliche Durchlaufzeit der einzelnen Aufträge minimiert (vgl. [GT95], S.358ff). Der Ablauf des Johnson- Algorithmus kann wie folgt beschrieben werden:

1. Für alle Aufträge i_1, \dots, i_n werden die zugehörigen Bearbeitungszeiten auf den beiden Maschinen A und B erfasst und in einer Tabelle dargestellt
2. Der Auftrag mit der geringsten Bearbeitungszeit $\min(A_i, B_i)$ wird gesucht.
3. Ist der Auftrag an Maschine A wird er dort an die erste Position gereiht
4. Ist der Auftrag an Maschine B wird er dort an die letzte Position gereiht
5. Der Auftrag wird aus der Liste gestrichen und es wird nach dem Auftrag mit der nun geringsten Bearbeitungszeit $\min(A_i, B_i)$ gesucht
6. Der Auftrag wird nach dem Auftrag/ Aufträgen auf Maschine A bzw. vor dem Auftrag/Aufträgen auf Maschine B eingereiht.
7. Die Schritte 4 und 5 werden solange wiederholt bis alle Aufträge zugeteilt sind.

Die beiden Auftragsfolgen auf Maschine A und B werden nun zusammengehängt und ergeben damit den optimalen Belegungsplan. Das ist möglich, da Aufträge im Zwei- Maschinenfall bei der selben Bearbeitungsfolge auf beiden Maschinen zumindest eine optimale Lösung aufweisen. Verschiedene Belegungsfolgen führen nur in Ausnahmefällen ebenfalls zu einer optimalen Lösung. In der Regel führen verschiedene Belegungsfolgen auf der zweiten Maschine zu zusätzlichen Wartezeiten und sind daher suboptimal.

Der Johnson- Algorithmus lässt sich auch auf stochastische Bearbeitungszeiten anwenden (vgl. [Nah05], S. 431ff). An Stelle der bekannten Fertigungszeiten wird dann mit den erwarteten Bearbeitungszeiten $E(A_j), E(B_j)$ gearbeitet. Die Minimierung der erwarteten Bearbeitungszeiten liefert dasselbe optimale Ergebnis wie bei deterministischen Problemstellungen (vgl. [Nah05], S. 431ff).

3.2.2 NEH- Heuristik

Eine weitere Lösungsmöglichkeit für die Reihenfolgeplanung bei Fließfertigung stellt die NEH- Heuristik dar (vgl. [GT95], S.360ff). Diese ist nach Nawaz/ Enscore/ Ham benannt die dieses Verfahren 1983 entwickelten. Diese Heuristik strebt eine Minimierung der Durchlaufzeit an. Der Rechenaufwand ist vergleichsweise gering und lässt sich bei entsprechender Implementierung weiter verringern. Die Lösungsqualität ist dabei vergleichbaren Heuristiken sogar überlegen. Sie ist auch für den Mehrmaschinenfall bei mehreren verschiedenen Produktionsaufträgen geeignet.

Für die Basislösung werden alle Fertigungsaufträge i_1, \dots, i_n entsprechend ihrer gesamten Bearbeitungszeit $\sum T_{ij}$ auf den Stationen j_1, \dots, j_m absteigend sortiert. Der erste Auftrag wird danach in die endgültige Lösung übernommen. Danach wird jeweils ein Arbeitsauftrag i zum endgültigen Plan ergänzt und in die bestehenden Fertigungsaufträge derart eingereiht, dass die relative Reihenfolge aller bisherigen Fertigungsaufträge i erhalten bleibt, dh. zugeordnete Fertigungsaufträge werden in ihrer Reihenfolge nicht mehr verändert.

3.2.3 Heuristik für ein Mehrmaschinen- Problem

Eine Erweiterung des Johnson- Algorithmus auf Probleme mit m - Maschinen stellten Cambell, Dudek und Smith 1970 vor ([CDS70]). Mit diesem Algorithmus ist es möglich, ein m - Maschinenproblem für $P = p_1, \dots, p_n$ Aufträge in ein Zwei- Maschinenproblem zu transferieren und dann zu lösen. Die Gesamtmenge aller Maschinen $M = \{m_1, \dots, m_m\}$ wird dabei in die beiden Teilmengen $m'_{A_i} = \sum_{i=1}^k m_{A_i}$ und $m'_{B_j} = \sum_{j=l}^m m_{B_j}$ geteilt. In $c = m - 1$ - Iterationsschritten kann dieses Problem nun gelöst werden. In Schritt $c = 1$ wird das erste Element aus $m'_{A_1} = A_1$ sowie das letzte Element aus $m'_{B_m} = B_1$ genommen. Mittels Johnson- Algorithmus wird dann für alle Aufträge gelöst, so als ob nur die beiden Maschinen A_1, A_2 zu durchlaufen wären. Im nächsten Schritt $c = 2$ wird die Maschine A_1 zur Pseudomaschine A_2 erweitert in dem die nächste Maschine hinzugenommen wird. Die Fertigungszeiten t_{Acp} der beiden einzelnen Maschinen werden dabei für die Fertigungszeit auf der Pseudomaschine addiert.

$$t_{Acp} = \sum_{i=1}^c t_{cp}$$

für alle $p \in P$

Die Maschine B_1 wird zur Pseudomaschine B_2 erweitert in dem die die Fertigungszeit der vorhergehende Maschine dazuaddiert wird.

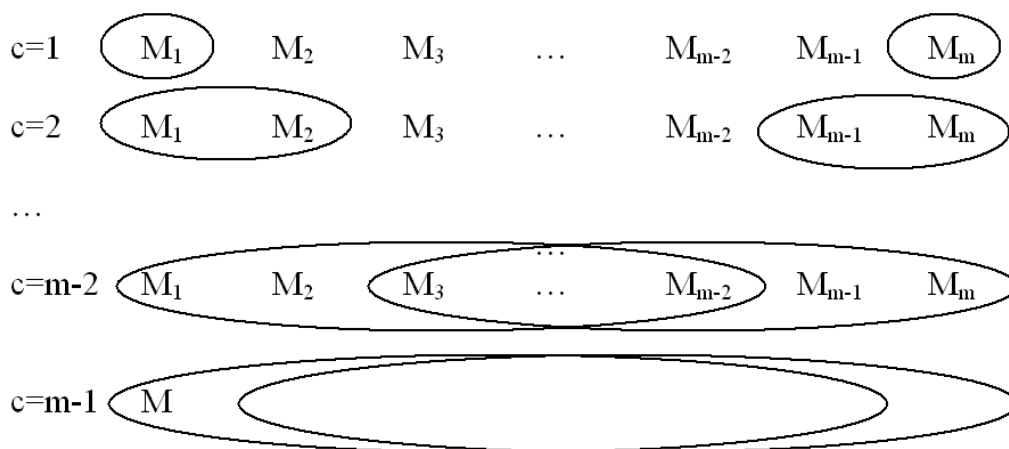


Abbildung 3.2: Iterationsstufen beim CDS- Algorithmus

$$t_{Bcp} = \sum_{j=m-c}^m t_{cp}$$

für alle $p \in P$

Nun wird wiederum mittels Johnson- Algorithmus für A_c, B_c gelöst. Die Pseudomaschinen werden nun schrittweise so lange erweitert, bis alle Maschinen verplant sind (siehe Abb. 3.2). Der Beste der c Einlastungspläne wird dann gewählt. Dieser kann dann in einem GANTT- Chart auch grafisch dargestellt werden.

Dieser Algorithmus ließe sich auch auf das (vereinfachte) Problem bei M-U-T anwenden. Dazu müssten folgende Annahmen getroffen werden:

- Die Bearbeitungszeit an allen Arbeitsstationen ist deterministisch. Dazu wird der Erwartungswert $E(t_i) = t_i$ angenommen
- Die Kapazität an allen Arbeitsstationen ist ausreichend und konstant
- Arbeitsstationen werden nacheinander in immer derselben Reihenfolge durchlaufen
- Die Anzahl der Aufträge ist bekannt
- Die Verteilung auf die einzelnen Varianten ist vor Produktionsbeginn vollständig bekannt

Bei der Bildung der Pseudomaschinen muss auf parallele Pfade geachtet werden. Dafür ist die CDS- Heuristik nicht ausgelegt. Weiters werden auch keine zirkularen Bezüge berücksichtigt wie das im Falle der Lackiererei zutrifft. Hier kommen die Aufträge in mehreren Fertigungsschritten auf dieselbe Maschine/ Arbeitsstation zurück und müssen eingelastet werden. Die Reihenfolge der Auftragseingänge muss ebenfalls schon bekannt sein.

Nachteilig ist bei allen Heuristiken, dass sie von jeweils leeren Systemen ausgehen. Das ist als Ausgangssituation allerdings unbefriedigend da normalerweise schon Aufträge im System laufen während die neuen Aufträge eingeplant werden. Weiters werden stochastische Fertigungszeiten nicht berücksichtigt und auch die Verfügbarkeit der Arbeitsstation wird nicht überprüft bzw. sichergestellt. Auch der Produktionsplan muss bereits bekannt sein. Das ist bei operativer Planung gegeben, nicht aber wenn der Zeithorizont des Modell auf die taktische Ebene erweitert wird. Weiters ist es auch nicht möglich bzw. sehr schwierig, Planungsheuristiken für die Zwecke der Fertigungsoptimierung zu verwenden. Diese sind primär für die konkrete Ablaufplanung in einer bekannten Produktionsumgebung gedacht. All diese Punkte sprechen dafür, das Modell mittels Methoden des Operations Research oder Simulation zu lösen.

Kapitel 4

Methoden des Operations Research

Operations Research (OR) bezeichnet einen Wissenszweig, der sich mit der Analyse von praxisnahen, komplexen Problemstellungen im Rahmen eines Planungsprozesses zum Zweck der Vorbereitung von möglichst guten Entscheidungen durch die Anwendung mathematischer Methoden beschäftigt. Die Hauptaufgaben des OR bestehen in der Abbildung eines realen Entscheidungsproblems durch ein Optimierungs- oder Simulationsmodell (...) und die Anwendung bzw. Entwicklung eines Algorithmus zur Lösung des Problems. Dabei spielt Softwareunterstützung eine zentrale Rolle. ([Dom07], S.1)

Operations Research arbeitet mit mathematischen Modellen. Ziel ist es, durch Optimierung ein definiertes Problem zu lösen. Je nach Problemstellung bietet OR verschiedene Methoden zur Lösung an:

- lineare Optimierung
- dynamische Optimierung
- nichtlineare Optimierung
- Netzplantechnik bzw. Kritische- Pfad Analysen
- Warteschlangenmodelle
- Simulation
- Spieltheorie

- Meta- Heuristiken

OR- Methoden finden auch in der Produktionsplanung Anwendung. Um den Ablauf der im Kapitel 3 beschriebenen Fertigungslayouts und Modelle planen, simulieren und umsetzen zu können gibt es Produktionsplanungsmodelle. Bei diesen Modellen geht es um die Mengenplanung und -optimierung sowie die Terminplanung und -steuerung der Produktion. Bei den heutzutage verwendeten Produktionsplanungs- und steuerungssystemen (PPS) sind diese auch mit den anderen Funktionsbereichen des Betriebes verbunden (Beschaffung, Lager, Rechnungswesen, Qualitätskontrolle etc.).

Produktionsplanungsmodelle werden sowohl direkt für die programmorientierte Bedarfsrechnung (Beschaffungsplanungsmodelle) als auch für Optimierungsrechnungen (Produktionsprogrammoptimierung) eingesetzt.

Die Modelle der Produktionsprogrammoptimierung bestehen zumeist aus einer linearen Zielfunktion und linearen Restriktionen und lassen sich mit Verfahren der linearen Optimierung (LP) lösen (vgl. [EI93a], S.1715).

Planungsmodelle der operativen Ebene greifen für diese Aufgabenstellung zu kurz. Mit einfachen Planungsinstrumenten wie Silver- Meal Verfahren oder Groff- Verfahren zur Losgrößenplanung oder Netzplantechniken zur Ressourceneinsatzplanung ist dieses Modell auch nicht mehr lösbar. Hierfür existiert ausreichend Literatur ([Nah05], [GT05]). Auch die Maschinenbelegungsplanung (vgl. [Neu02], S.474ff) lässt sich nicht anwenden. Die Frage bei Maschinenbelegung ist, wann welche Aufträge auf welcher Maschine bearbeitet werden sollen. Das setzt deterministisch bekannte Aufträge voraus. Da die Aufträge in diesem Beispiel stochastisch generiert werden scheiden die Verfahren der operativen Ebene aus.

Mittlerweile sind praktisch in allen verarbeitenden Betrieben PPS- Systeme im Einsatz. Diese sind den FFS- Systemen und CNC- Bearbeitungszentren übergeordnet und ermöglichen durch vernetzte und komplexe Programme eine optimale Fertigungssteuerung.

Auch die Methoden der taktischen Planung (Gozintograph, Dispositionsstufenverfahren, etc.) sind nicht verwendbar. Diese setzen ebenfalls deterministische oder bekannte Daten voraus.

Komplexere Modelle zur Leistungsanalyse von FFS- Systemen (dynamische Optimierung, Warteschlangentheorie) können auch nicht angewendet

werden. Das klassische CQN- Modell (Closed Queuing Network- Modell), welches Tempelmeier [TK93] ausführlich behandelt, lässt sich nur auf definierte, genau abgegrenzte FFS- Systeme anwenden. Er geht auch immer davon aus, dass die zu produzierenden Produkte kleiner sind und auf Paletten, die als Puffer dienen, gemeinsam gelagert und transportiert werden können. Das ist in diesem Fall, wo komplette Nutzfahrzeuge aufgebaut werden und alle verbauten Halbfabrikate entsprechend groß sind, nicht möglich.

Hier sind die Daten jedoch durchwegs stochastisch. Damit scheiden auch Verfahren mit linearen Gleichungssystemen aus, obwohl LP- Verfahren des Operations Research in der Praxis einfach zu handhaben sind und auch die Lösung größerer Aufgabenstellungen auf Grund leistungsfähiger Programme möglich ist. Das vorliegende, nicht lineare stochastische Problem wird deshalb mit Hilfe von Simulation gelöst.

Einige der bekannten Methoden des OR werden kurz vorgestellt um einen Vergleich zur Simulationsmethode ziehen zu können. Damit soll die Abgrenzung in den Anwendungsgebieten der verschiedenen Methoden dargestellt sowie Gemeinsamkeiten und Unterschiede hervorgehoben werden.

4.1 Modellierungsschritte

Die verschiedenen Verfahren zur mathematischen Lösung von Problemstellungen arbeiten mit unterschiedlichen Lösungsmethoden. Diese Methoden sind meistens auch schon in geeigneter Software implementiert. Das erleichtert den Benutzern die Arbeit und setzt auch kein besonderes Methodenverständnis voraus.

Die Berechnung am Computer ist aber nur ein Teilschritt im Lösungsprozess. Ebenso wichtig sind alle anderen Schritte davor und danach. Diese sind unabhängig vom Lösungsverfahren anzuwenden und lassen sich grob skizzieren (vgl. [HL04], S.8):

- Definierung des Problems und Sammlung der relevanten Daten
- Formulierung eines mathematischen Modells um das Problem darzustellen
- Entwicklung eines computerbasierten Lösungsmodells um Lösungen des Problems am Modell zu bekommen
- Testen des Modells und notwendige Korrekturen durchführen

- Vorbereitungen für die Anwendung des Modells zum Einsatz im Unternehmen
- Implementierung und Umsetzung

Die Situationsanalyse stellt eine IST- Analyse dar und soll die Ausgangssituation exakt erklären. Sie bietet die Basis für die Formulierung einer konkreten Aufgabenstellung. Die Analyseergebnisse werden als Quelle für die Konkretisierung der Ziele sowie als Inputgrößen für die Modellierung verwendet. Oftmals ist es schwierig das Problem in konkrete, messbare Ziele zu fassen. Auch die korrekte Formulierung der Beschränkungen und der Beziehungen der Zielgrößen und Einflussfaktoren untereinander kann schwierig werden. In dieser Phase muss trotz aller Probleme genau gearbeitet und entsprechend Zeit aufgewendet werden.

Dabei muss zwischen dem Hauptziel sowie den konkret heruntergebrochenen Teilzielen unterschieden werden. Gerade diese Teilziele zu identifizieren ist entscheidend für den Erfolg der Simulation. Die Abhängigkeiten zwischen den Zielen zu erkennen und am Modell umzusetzen ist ein wesentlicher Schritt einer guten Modellierung. Speziell bei konkurrierenden und divergierenden Teilzielen ist das eine große Herausforderung. Die Erfüllung sämtlicher Teilziele ermöglicht die Erreichung des Hauptziels.

Es ist wichtig, alle Entscheidungsträger und von der Aufgabenstellung betroffenen Mitarbeiter einzubeziehen. Damit wird ein vollständiger Überblick und die Sicht von allen Seiten sichergestellt. Für ein ungenau oder falsch definiertes Problem wird man auch nur ungenaue oder falsche Lösungen bekommen. Später am Modell noch vereinfachende Annahmen zu treffen ist besser als von vornherein ungenau an das Problem heranzugehen.

Folgende Fragen können bei der Situationsanalyse und Zielformulierung hilfreich sein (vgl. [HL04], S.955):

- Welche Problemstellung will der Auftraggeber gelöst haben?
- Was ist die generelle Zielsetzung des Modells?
- Welche speziellen Probleme sollen gelöst werden?
- Welche alternativen Konfigurationen des Modells sollen berücksichtigt werden?
- Welche Kennzahlen des Modells sollen gemessen werden, um die Zielerreichung zu prüfen?

- Welcher zeitliche Rahmen ist bis zur Fertigstellung vorgesehen?

Nach der Situationsanalyse und Zielformulierung muss das Problem nun in eine mathematische Form gebracht werden. Diese mathematische Formulierung besteht normalerweise aus Entscheidungsvariablen, einer Zielfunktion und Nebenbedingungen sowie definierten Parametern. Bereits hier muss man sich für eine OR- Methode entscheiden um das Problem entsprechend mathematisch formulieren zu können. Hier ist die Kenntnis über die Methoden des OR sowie deren Anwendung unerlässlich. Die Wahl einer ungeeigneten Methode wird keine brauchbaren Ergebnisse liefern. Dabei ist wichtig die weniger präzise verbale Situationsanalyse nun in ein mathematisch exaktes Modell zu bringen, besonders in Hinsicht auf die korrekte Formulierung der Zielfunktion. Bei mehreren Entscheidungsvariablen ist es auch wichtig, die Abhängigkeiten voneinander zu erkennen und zu berücksichtigen. Auch bei divergierenden Teilzielen muss die Zielfunktion so formuliert werden, dass das Hauptziel entsprechend optimiert werden kann.

Anschließend an die mathematische Formulierung kann die Datenerhebung begonnen werden. Oftmals passiert dieser Schritt auch parallel zur mathematischen Formulierung. Hier sollte man zielgerichtet die richtigen Daten erheben. Oftmals stehen eine Fülle von Daten zur Verfügung aus denen nun die relevanten Daten gefunden und herausgefiltert werden müssen.

Dafür gibt es allgemeine Anforderungen an Daten die auf jeden Fall erfüllt werden sollen. Kosturiak hat diese Anforderungen für den Fall der Simulation formuliert, diese können aber auf alle Methoden des OR übertragen werden (vgl. [KG95], S.110):

- Die Daten müssen **vollständig** erhoben werden, dh. alle Modellkomponenten und Schnittstellen müssen datenmäßig versorgt werden.
- Die Daten müssen der festgelegten **Modellgenauigkeit** entsprechen.
- Die Daten müssen eine **repräsentative Stichprobe** der zu untersuchenden Verhältnisse darstellen.
- Die Daten müssen **von zufälligen Schwankungen**, die nicht typisch für einen Untersuchungsfall sind, **gereinigt** werden.
- Die Daten müssen in **ausreichendem Umfang** verfügbar sein, so dass der Datenvorrat für den gesamten Verlauf des Experiments ausreicht.
- Die Daten müssen **reproduzierbar** sein, sodass man verschiedene Modellvarianten von gleichen Datenreihen entwickeln kann.

- Die Daten sollten so aufbereitet werden, dass sie **direkt** in das Modell **übernommen** werden können, und sie müssen den **Anforderungen des Modells entsprechen**.

Die anschließende Testphase mit Verifizierung und Validierung stellt einen wesentlichen Meilenstein vor dem Durchführen der Experimente dar. Die **Modellverifikation** ist eine logische Überprüfung der inneren Logik des Simulationsmodells ([KG95], S.123). Hier wird überprüft, ob das Modell in sich selbst konsistent und lauffähig ist. Es wird auch nach logischen Fehlern im System gesucht. Besonders fehlende und falsche Verknüpfungen und falsche Parameterübergabe, fehlende Komponenten etc. sollen in diesem Schritt erkannt und korrigiert werden. Die Verifikation erfolgt unabhängig davon, WAS mit dem Modell erreicht werden soll sondern behandelt nur das WIE des Modells. Die **Validierung** ist die Überprüfung der hinreichenden Übereinstimmung von Modell und Originalsystem. Die Validierung erfasst also die inhaltliche Richtigkeit des Modells. Hier geht es darum, einen Vergleich mit der gewählten Aufgabenstellung und Zielsetzung herzustellen. Hier wird das WAS geklärt: Was macht mein Modell und entspricht dies dem Originalsystem?

Ist das Modell nun getestet und korrekt kann mit der Lösungssuche begonnen werden. Bei exakten Verfahren benötigt es dazu nur einen Modelldurchlauf um die optimale Lösung zu finden. Es kann aber auch sein, dass mehrere Lösungen optimal sind. Hier bietet die gewählte Software dann die verschiedenen Lösungen an oder der Modellierer muss selbst alle möglichen Varianten durchgehen. Bei heuristischen Verfahren oder der Simulation muss das Modell mehrmals durchlaufen werden um aussagekräftige und brauchbare Ergebnisse für die folgende Auswertung zu erhalten. Da diese keine optimale Lösung liefern muss öfters getestet und berechnet werden um sich an die optimale Lösung anzunähern.

Am Ende der Modellierung wird die Lösung analysiert und interpretiert. Die Interpretation der ausgewerteten Ergebnisse erfordert Erfahrung mit dem realen System. Eine falsche Interpretation der Daten und falsche Schlussfolgerungen für das reale System können weiters ein bislang richtiges Modell und korrekte Daten als fehlerhaft dastehen lassen. Gerade die Interpretation entscheidet über Erfolg und Misserfolg des Modells.

4.2 Lineare Optimierung

Die Modelle der linearen Optimierung bestehen aus einer linearen Zielfunktion und einer Vielzahl linearer Restriktionen. Zahlreiche Unternehmungsmodelle, Absatzplanungsmodelle und Transportmodelle, Finanzplanungsmodelle und Produktionsplanungsmodelle sind vom Typ der linearen Optimierung. Die größte Bedeutung hat die lineare Optimierung für die Fertigungsplanung. [EI93a], S. 1322.

Lineare Optimierungsmodelle besitzen in ihrer Grundform eine lineare Zielfunktion Z der Form

$$Z = \sum c_i x_i \rightarrow \max!$$

Zielfunktionen der linearen Programmierung (LP) werden immer als Maximierungsaufgabe formuliert. Sie haben beliebig viele Entscheidungsvariablen x_i , $i \in \{1, \dots, n\}$ die durch die Faktoren c_i bewertet werden.

LP- Modelle können auch beliebig viele Nebenbedingungen $j \in \{1, \dots, m\}$ enthalten. Nebenbedingungen sind ebenfalls lineare Gleichungen der Form

$$\begin{aligned} \sum a_{i,j} x_i &\leq b_j & j \in \{1, \dots, l\} \\ \sum a_{i,k} x_i &\geq b_k & k \in \{m, \dots, n\} \end{aligned}$$

Da die Zielfunktion maximiert wird, muss mindestens eine Nebenbedingung eine obere Restriktionsgrenze darstellen. Jede Entscheidungsvariable x_i wird durch einen Limitierungsfaktor a_{ij} beschränkt.

Weiters gilt für alle Entscheidungsvariablen die Bedingung

$$x_i \geq 0$$

Eine Lösung wird als zulässig bezeichnet, wenn alle Nebenbedingungen erfüllt sind. Eine Lösung wird als unzulässig bezeichnet, wenn mindestens eine Nebenbedingung nicht erfüllt ist. Eine optimale Lösung bietet den größten Zielfunktionswert bei gleichzeitiger Einhaltung aller Nebenbedingungen (vgl. [HL04], S.35). Einfache lineare Probleme können auch graphisch dargestellt werden. Hier ist einfach ersichtlich, dass die optimale Lösung an zumindest einem Eckpunkt liegen muss (siehe Abb. 4.1). Dieser Eckpunkt wird von den limitierenden Nebenbedingungen gebildet. Das sind jene Bedingungen, die für die jeweilige Zielfunktion den maximalen Wert darstellen ohne den zulässigen Lösungsraum zu verlassen. Es ist auch möglich, dass LP- Modelle keine Lösung besitzen. Die Nebenbedingungen sind nicht einzuhalten und

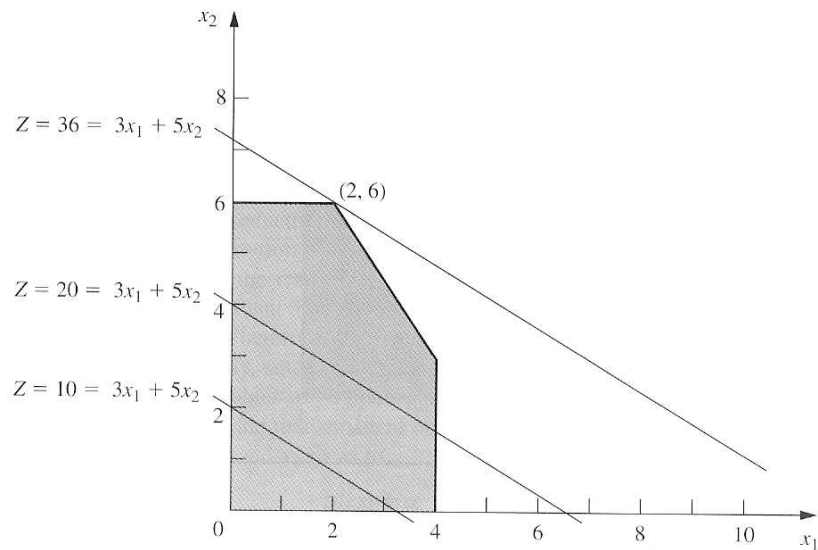


Abbildung 4.1: Graphische Lösung eines LP- Problems (vgl. [HL04], S.32)

müssen ggf. gelockert werden. Auf der anderen Seite ist es auch möglich, beliebig viele optimale Lösungen für ein Problem zu erhalten. Das ist der Fall, wenn die Zielfunktion dieselbe Steigung wie eine der Nebenbedingungen aufweist. In der graphischen Darstellung würde die Zielfunktion dann parallel zu einer der Nebenbedingungen verlaufen und alle Punkte auf dieser Nebenbedingung stellen optimale Lösungen dar. Durch entsprechende Formulierungen können beinahe alle linearen Problemstellungen durch LP gelöst werden. Die graphische Lösung ist nur für Problemstellungen mit zwei Variablen x_1, x_2 durchführbar und dient primär zum Verständnis der LP- Methode.

Eines der bekanntesten Verfahren zur Lösung realer Probleme ist das Simplex- Verfahren welches George. B. Dantzig 1947 entwickelte. Diese Methode basiert auf der Matrizenrechnung. Damit lassen sich auch große LP's mittels Rechneinsatz rasch lösen. Ausgehend von einer zulässigen Lösung (meistens $x_i = 0$) werden zulässige Eckpunkte des Lösungsraumes nacheinander abgesucht. Bei der Simplexmethode wird ausgehend vom aktuellen Eckpunkt immer die nächstgelegene beste Lösung gewählt bis das Optimum erreicht ist. Mit wenigen Iterationsschritten lässt sich damit die optimale Lösung auch bei höherdimensionalen Problemen rasch finden. Der Zielfunktionswert Z wird über die beiden Vektoren c und x berechnet.

$$Z = cx \rightarrow \max!$$

Die Nebenbedingungen werden durch die Matrix A und den Vektor b

gebildet.

$$Ax \leq b$$

Wie bei allen LP- Modellen gilt

$$x \geq 0$$

wobei c der Reihenvektor der Faktoren ist

$$c = [c_1, c_2, \dots, c_n]$$

x, b und 0 sind Spaltenvektoren

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad 0 = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

und A die Matrix der Limitierungsfaktoren a_{ij}

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Für die rechnergestützte Lösung muss dieses Problem nochmals umformuliert und um die Schlupfvariablen x_s und die Identitätsmatrix I erweitert werden. Die Nebenbedingungen haben dann die Form

$$[A, I] \begin{bmatrix} x \\ x_s \end{bmatrix} = b \quad \text{und} \quad \begin{bmatrix} x \\ x_s \end{bmatrix} \geq 0$$

Für eine genaue Erklärung zur Simplex- Methode siehe Hillier/ Lieberman ([HL04], Kap. 4,5).

Exakte Methoden wie die LP- Modellierung haben sich als Lösung für die Fertigungsplanung vor allem bei kurzfristiger Planung und in der Serien- und Massenfertigung durchgesetzt. Bei der operativen Planung sind die Produktionsmengen und Kapazitäten bekannt. Die kurzfristige Planung berücksichtigt keine stochastischen Einflüsse wie schwankende Kapazitäten, Bearbeitungszeiten oder Maschinenausfälle. Diese wären mit LP nicht lösbar, werden aber bei mittelfristiger Planung sehr wohl berücksichtigt.

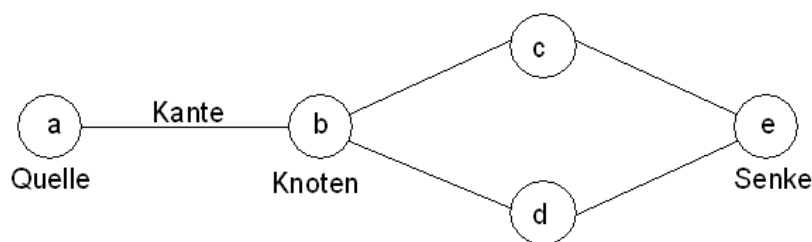


Abbildung 4.2: Einfacher Netzplan

4.3 Netzplantechnik

Netzplantechniken eignen sich besonders für die Planung von Projekten, finden aber auch Einsatz bei logistischen Problemen. Basis für die Erstellung und Bearbeitung von Netzplänen ist die zugrunde liegende Graphentheorie. Alle Netzpläne sind aus Graphen aufgebaut. Ein Graph besteht, wie in Abb. 4.2 gezeigt, aus Knoten und Kanten. Die Kanten verbinden die einzelnen Knoten eines Netzwerkes. Der Startknoten a wird als Quelle, der Zielknoten e als Senke bezeichnet. Es kann in einem Netzplan auch mehrere Quellen und Senken geben. Ziel der Netzplantechnik ist es, den kritischen Pfad durch das Netzwerk zu ermitteln und zu optimieren um die Durchlaufzeit zu minimieren. Der längste Weg durch das Netzwerk (dh. der Weg mit der längsten Durchlaufzeit) wird dabei als kritischer Pfad bezeichnet. Entsprechend wird die Netzplantechnik oft auch als **Kritische-Pfad-Analyse** bezeichnet.

Mit der Netzplantechnik lassen sich auch große praktische Probleme (etwa Verkehrsnetze mit hunderten von Straßen oder Versorgungsnetzwerke mit Millionen von Kunden) mit relativ geringem Rechenaufwand exakt lösen (vgl. [Neu02], S. 176). Dabei kann zwischen grundlegenden netzwerkbasieren OR-Methoden unterschieden werden (vgl. [HL04], S. 414, [Dom07], S.97). Diese wurden in den späten 1950er Jahren unabhängig voneinander entwickelt:

- CPM (critical path method)
- MPM (Metra Potential method)
- PERT (program evaluation and review technique)
- GERT (graphical evaluation and review technique)

Die drei erwähnten Netzplantechnik-Methoden CPM, PERT und MPM sind - im wesentlichen unabhängig voneinander- gegen Ende der fünfziger Jahre entwickelt worden. Neuere Verfahren der

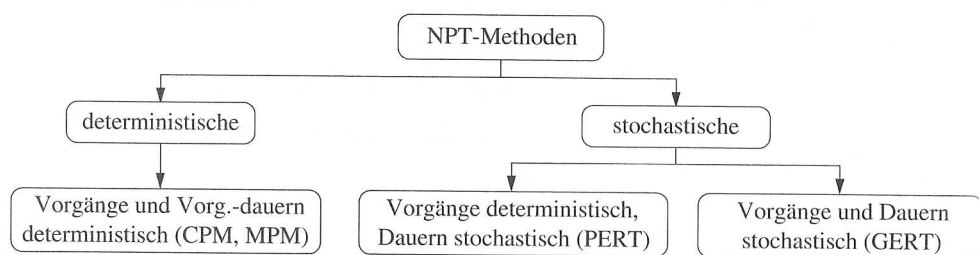


Abbildung 4.3: Arten von Netzplänen ([Dom07], S.99)

Netzplantechnik enthalten in der Regel Elemente von mehr als einer der drei ursprünglichen Methoden gemeinsam ([Neu75]).

Diese Verfahren lassen sich wie in Abb. 4.3 dargestellt in deterministische und stochastische Verfahren unterteilen. CPM verwendet ein Vorgangspfeilnetz mit Ende- Start- Verknüpfung der Vorgänge, während MPM auf einem Vorgangsknotennetz mit Start-Start-Beziehung basiert [Neu02], S.230). Mittlerweile sind auch kombinierte Verfahren im Einsatz, z.B. die PERT/CPM-Technik. Zum allgemeinen Verständnis von Netzplänen wird die CPM- Methode kurz vorgestellt, die PERT- Methode soll hinsichtlich der Verwendbarkeit für diese Arbeit diskutiert werden.

4.3.1 CPM- Netzpläne

GANTT- Charts und Gozintographen sind einfache Anwendungsbeispiele der CPM-Netzplantechnik die auch in der Produktionsplanung Verwendung finden (siehe [GT05], 179ff).

CPM- Netzpläne besitzen immer genau eine Quelle und eine Senke. Die Senke ist über die Knoten von der Quelle aus erreichbar. Ein CPM- Netzplan ist ferner schwach zusammenhängend und zyklensfrei [Neu02], S. 233). CPM-Netzpläne bestehen aus Knoten der Knotenmenge $V = \{v_1, \dots, v_n\}$ und der Pfeilmenge $E = \{e_1, \dots, e_n\}$ welche die Kanten des Netzplanes enthalten. Ein Vorgang stellt die Verbindung zweier Knoten der Knotenmenge V mit einer Kante der Pfeilmenge E dar. Vorgänge werden durch die Indexmenge

$$I := \{1, \dots, n\}, \quad J := \{(i, j) \in I \times I : [v_i, v_j] \in E\}$$

beschrieben. Die Dauer eines Vorganges $[v_i, v_j]$ wird mit $D_{ij}((i, j) \in J)$ bezeichnet. Damit lassen sich alle Wege durch den Netzplan beschreiben. Einer dieser Wege ist der kritische Pfad. Der kritische Pfad ist der längste Weg

von der Quelle zur Senke und weist keine Pufferzeit auf. Jeder Vorgang $v_{i,j}$ auf dem kritischen Pfad wird als kritischer Vorgang bezeichnet. Da für die Abarbeitung eines Netzplanes alle Vorgänge $v_{i,j}$ erledigt werden müssen, entspricht die kürzeste Gesamtdauer des Projekts dem kritischen Pfad.

Alle Vorgänge $v_{i,j}$ in einem CPM- Netzplan sind deterministisch bekannt. Mittels CPM- Netzplantechnik lässt sich also eine optimale Lösung bestimmen. (vgl. [Neu75], S.196

Für die Projektplanung müssen also die vier Größen

- Kürzeste Dauer des gesamten Projekts
- Kritische Vorgänge
- Anfangs- und Endtermine aller Vorgänge
- Pufferzeiten aller Vorgänge

bestimmt werden. Dazu werden für alle Vorgänge $v_{i,j}$ die Größen früheste Anfangszeit ($FAZ_{i,j}$), früheste Endzeit ($FEZ_{i,j}$), späteste Anfangszeit ($SAZ_{i,j}$) und späteste Endzeit ($SEZ_{i,j}$) bestimmt sowie die Pufferzeit $P_{i,j}$ bestimmt. Für jeden Vorgang $v_{i,j}$ wird weiters die Vorgangsdauer $D_{i,j}$ deterministisch festgelegt. Durch Vorwärtsterminierung kann vom Projektstarttermin T_0 beginnend $FAZ_{i,j}$ und $FEZ_{i,j}$ sowie der Projektendtermin T_n bestimmt werden. Dieser entspricht dem frühest möglichen Endtermin der Senke v_n und wird mit dem spätest möglichen Endtermin gleichgesetzt, dh.

$$T_n := FEZ_n := SEZ_n$$

Durch Rückwärtsterminierung kann nun für jeden Vorgang $SAZ_{i,j}$ und $SEZ_{i,j}$ sowie die Pufferzeit $P_{i,j}$ bestimmt werden. Durch das Netzwerk gibt es somit mindestens einen Weg, bei dem die Summe der Pufferzeiten $\sum P_{i,j} = 0$ gilt. Das ist der kritische Pfad durch den Netzplan.

Netzpläne werden oft auch graphisch dargestellt. Die Notation bei CPM- Netzplänen werden dabei wie in Abb. 4.2 dargestellt gewählt. Die Knoten enthalten die relevanten Informationen für jeden Vorgang. Die Kanten zeigen den Weg durch den Netzplan. Für eine genauere Beschreibung der CPM- Methode siehe [Neu75], S.199ff, [Neu02], S.234ff, [LG91], S.44ff, [Nah05], S.485ff.

4.3.2 PERT- Netzpläne

PERT- Netzwerke wurden für Anwendungsfälle entwickelt, bei denen die Vorgangsdauern nicht deterministisch bekannt sind. Dabei steht die Kontrolle

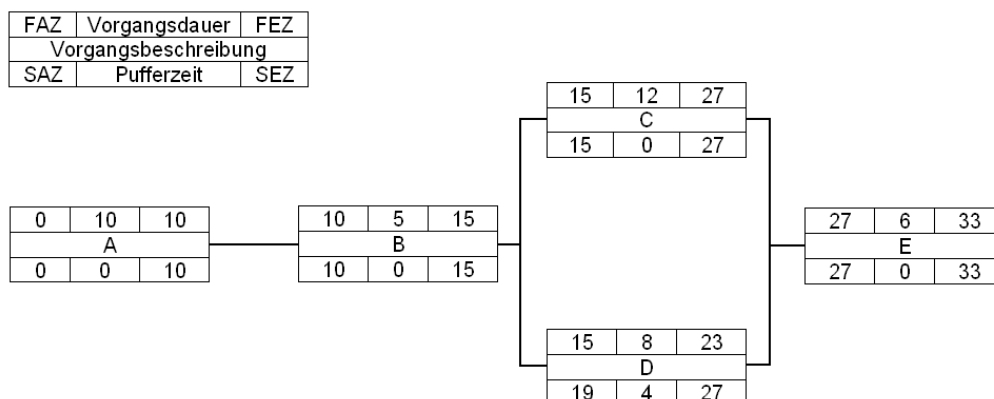


Abbildung 4.4: Einfacher CPM- Netzplan

und Analyse des Projektfortschrittes im Vordergrund und nicht die Steuerung der Arbeitsvorgänge im Projekt (vgl. [Neu75], S.210). Die Vorgangsdauern in PERT- Netzplänen sind deshalb β - verteilt. Die Vorgangsdauer D setzt sich dabei aus den drei Zeitschätzungen a (optimistischer Schätzwert), b (pessimistischer Schätzwert) und m (häufigster Vorgangswert) zusammen (vgl. [LG91], S.41, [Neu02], S.366ff). Diese werden zur erwarteten Vorgangszeit t_e zusammengesetzt und man erhält

$$t_e = \frac{a + b + 4m}{6}$$

Für die Varianz (Streuung) der Vorgangsdauern wird folgende Näherungsformel verwendet:

$$V = \frac{(a - b)^2}{36}$$

Damit stellen die frühesten Eintrittstermine (FEZ) und spätesten Eintrittstermine (SEZ) für jeden Knoten i stochastische Zufallsgrößen dar. Die Erwartungswerte MFZ_i und MSZ_i dieser Zufallsgrößen bestimmt man ähnlich wie bei CPM.

Ein Weg ist dann kritisch, wenn sämtliche auf seinem Weg von der Quelle zur Senke liegenden Knoten kritische Ereignisse darstellen. Im Gegensatz zu CPM wird bei PERT ein kritischer Weg als Folge kritischer Ereignisse angesehen und nicht als Folge kritischer Vorgänge (vgl. [Neu75], S.221).

Kritik an PERT- Netzplänen richtet sich in erster Linie gegen die angenommene Betaverteilung für die Vorgangsdauern. In der Praxis würden auch viele andere Verteilungen, wie z.B. die wesentlich einfachere Dreiecksverteilung, genügen. Damit kann es auch zu sehr großen Fehlern im Vergleich zur

wahren Verteilung der Vorgangsdauern kommen. Weiters wird die vollkommene Unabhängigkeit der einzelnen Vorgangsdauern angenommen. Das ist in der Praxis oft nicht zutreffend da z.B. der gleiche Bearbeiter verschiedene Vorgänge bearbeitet. Besonders bei "seriell" aufgebauten Plänen, bei denen die Vorgänge größtenteils nacheinander ausgeführt werden, können sich die Fehler addieren. Die erwartete kürzeste Projektdauer wird oft als zu kurz angenommen. Als Varianz wird bei PERT die Summe der Varianzen der einzelnen Vorgänge am Weg festgelegt. Das ist nur dann zutreffend wenn die Vorgänge wirklich unkorreliert wären. Hier kann es in Einzelfällen zu Fehlern von 25% und mehr kommen. Die PERT- Methode sollte deshalb nur benutzt werden wenn CPM nicht verwendet werden kann. (vgl. [Neu75], S.231ff, [Neu02], S.367).

Auf Grund dieser Kritikpunkte wird die Netzplantechnik für dieses Modell nicht verwendet. Die Daten können im Modell nicht deterministisch bezogen sondern nur stochastisch festgelegt werden. Damit scheidet das CPM-Verfahren von vornherein aus. Die Datenauswertung in Kapitel 5 zeigt, dass eine Dreiecksverteilung für die Vorgangsdauern die beste Annäherung an die wahren Werte bietet. Weiters ist der Pfad durch den Montageprozess seriell und sehr lang (siehe Tab. 2.3) was zu entsprechend großen Fehlern bei Anwendung der PERT- Methode führen kann.

4.4 Warteschlangentheorie

Warteschlange treten an Bedienungssystemen auf, bei denen Ankunfts- und Abfertigungsprozess nicht durch zentrale Steuerung miteinander synchronisiert sind. (...) Die Warteschlangentheorie dient der Untersuchung der Entwicklung von Warteschlangen in Abhängigkeit von (a) der Ankunft der zu bedienenden Einheiten, (b) der Abfertigung (Bedienung) dieser Einheiten und (c) dem Bedienungssystem. Insbesondere interessieren die durchschnittlichen Wartezeiten (und die statistische Verteilung der Wartezeiten) sowie die durchschnittliche Länge der Warteschlange (und die statistische Verteilung der Warteschlange). [EI93a], S.2319

Nach D.G. Kendall werden Warteschlangenprozesse durch die drei Merkmale

- Ankunftsprozess

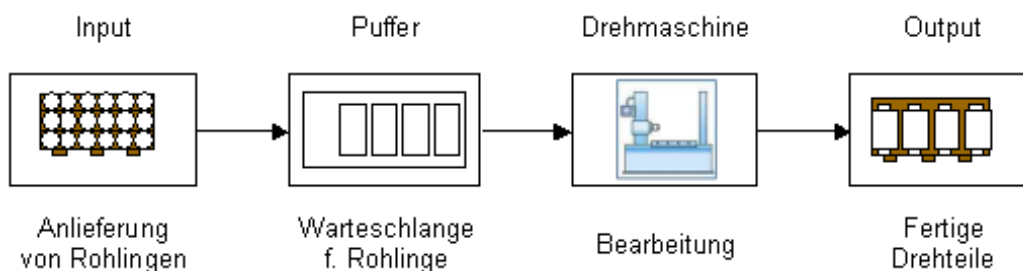


Abbildung 4.5: Einfaches Warteschlangensystem

- Abfertigungsprozess (bestehend aus Verweil- und Bedienprozess) und
- Anzahl der Abfertigungsschalter

spezifiziert.

Warteschlangenprozesse enthalten

- bewegliche Elemente, die in der Quelle generiert werden
- Warteschlangen, die die beweglichen Modelle aufnehmen und in denen sie auf ihre Bedienung warten
- Abfertigungsschalter, die die wartenden Objekte bedienen

Die Quelle enthält endlich oder unendlich viele beweglichen Elemente. Diese werden an Hand von Verteilungen oder deterministisch erzeugt. Häufig wird eine Poisson-Verteilung für die Anzahl und Verteilung der Elemente sowie exponentiell verteilte Zwischenankunftszeiten angenommen. In der Warteschlange findet der Verweilprozess statt. Auch Warteschlangen können endlich oder unendlich sein. Am Abfertigungsschalter wird der Bedienprozess durchgeführt. An Hand von Prioritätsregeln wird festgelegt, welches Element der Warteschlange als nächstes zur Bedienung gelangt und wie diese auf die (parallelen) Abfertigungsschalter verteilt werden. Der prinzipielle Aufbau eines Warteschlangensystems ist in Abb. 4.5 dargestellt.

Warteschlangensysteme eignen sich besonders für einfache Anwendungsfälle die genau diesem Schema entsprechen. Beispiele für Warteschlangensysteme sind in 4.1 dargestellt. In einem Warteschlangensystem muss das zu bearbeitende Element nicht unbedingt physisch beweglich sein. Beispielsweise können Maschinen defekt werden und dann auf die Reparatur durch einen Mechaniker warten. In diesem Fall kommt der Abfertigungsschalter zum Element. Dies ist für die mathematische Lösung von Warteschlangenproblemen jedoch irrelevant.

Prozess	System	Element	Abfertigungsschalter
Waren bezahlen	Supermarkt	Kunde	Kassa
Geld beheben	Bank	Kunde	Bankomat
Patient untersuchen	Krankenhaus	Patient	Arzt
Warentransport	Firmengelände	Paletten	Stapler
Maschine reparieren	Fabrikshalle	defekte Maschine	Mechaniker

Tabelle 4.1: Beispiele für Warteschlangensysteme

Interessante Kenngrößen von Wartesystemen sind (vgl. [Dom07], S.212):

- die durchschnittliche Systemlänge (Anzahl der Elemente in der Warteschlange und am Abfertigungsschalter)
- die durchschnittliche Schlängellänge (Anzahl der Elemente in der Warteschlange)
- die durchschnittliche Verweilzeit im System
- die durchschnittliche Wartezeit in der Warteschlange

Für die Klassifizierung und Unterscheidung der verschiedenen möglichen Problemstellungen in der Warteschlangentheorie wurde in der Literatur ein Klassifizierungscode entwickelt. Dieser Code kann sechsstellig sein nach dem Muster a/b/c/d/e/f, wird zumeist aber nur dreistellig (a/b/c) verwendet. Deshalb werden auch nur die ersten drei Codestellen kurz erklärt. Die sechs Stellen des Codes werden jeweils durch einen Schrägstrich getrennt. Die möglichen Codesegmente für jede Stelle sind definiert (vgl. [HL04], S.769, [Dom07], S.223ff):

- a: Beschreibung des Ankunftsprozesses
 - M bedeutet, dass die Ankünfte poissonverteilt und die Zwischenankunftszeit exponentialverteilt ist. Der Ankunftsprozess ist eine Markovkette
 - G steht für eine beliebige Verteilung des Ankunftsprozesses
 - D steht für deterministische, dh. fest vorgegebene Ankunftszeitpunkte.
 - E steht für die Erlangverteilung
- b: Die zweite Stelle charakterisiert den Abfertigungsprozess. Hier können dieselben Eintragungen wie beim Ankunftsprozess stehen.

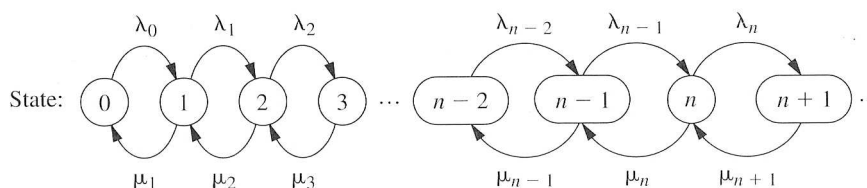


Abbildung 4.6: M/M/1- Warteschlangensystem ([HL04], S.781)

- Die dritte Stelle gibt an, wieviele Elemente parallel bearbeitet werden können, dh. wieviele Abfertigungsschalter im System sind.

4.4.1 Das Wartesystem M/M/1

Als einfachstes Modell von Warteschlangenprozessen dient der M/M/1- Prozess (Abb. 4.6) Der Ankunfts- und Abfertigungsprozess ist dabei jeweils poissonverteilt mit exponentialverteilten Zwischenankunftszeiten. Außerdem gibt es nur eine Warteschlange und einen Abfertigungsschalter. Die Zeitspannen zwischen zwei aufeinanderfolgenden Ereignissen, dh. die Zwischenankunftszeiten Z_n sowie die Bedienzeiten S_n , sind voneinander unabhängige Zufallsgrößen.

Weiters gelten folgende Notationen:

- $N(t)$: Anzahl der Elemente im Wartesystem zum Zeitpunkt t
- $P_n(t)$: Wahrscheinlichkeit, exakt n Elemente zum Zeitpunkt t im Wartesystem zu haben
- λ_n : Ankunftsrate für neue Elemente wenn bereits n Elemente im System sind. Im M/M/1- System ist die Ankunftsrate λ für alle Elemente n identisch, daher gilt: $\lambda_n = \lambda$
- μ_n : Servicerate für das System, wenn n Elemente im System sind. Im M/M/1- System ist die Servicerate μ für alle Elemente n identisch, daher gilt: $\mu_n = \mu$

Weiters wird angenommen, dass das Wartesystem zum Zeitpunkt 0 leer ist und die Elemente zu den Zeitpunkten T_1, T_2, \dots, T_n einzeln am Abfertigungsschalter ankommen und sich in die Warteschlange einreihen. Mit diesen Annahmen können nun die Kenngrößen des Wartesystems berechnet werden.

Der Auslastungsgrad ρ des Abfertigungsschalter beträgt

$$\rho = \lambda/\mu$$

Die Gesamtzeit des Elements n im System ist die Summe aus Wartezeit und Bedienzeit und wird auch Verweilzeit genannt.

$$W_n = W_nq + S_n$$

Die erwartete Anzahl Elemente im Wartesystem ist

$$L = \sum_{n=0}^{\infty} (nP_n)$$

Im Falle nur einer Abfertigungsstation ist die erwartete Anzahl der Elemente im System gleich mit der erwarteten Warteschlangenlänge

$$L = L_q$$

und bei stabilen Wartesystemen gilt weiters die Formel von Little mit

$$L = \lambda * W$$

Damit gilt weiters

$$L_q = \lambda * W_q$$

und für die durchschnittliche Wartezeit eines Elementes

$$W = W_q + \frac{1}{\mu}$$

Dieser Zusammenhang zwischen den Größen L , W , L_q und W_q ist besonders nützlich. Sobald eine dieser Größen durch Messung, Versuche, Datenauswertung etc. bestimmt werden kann können sofort auch alle anderen Größen berechnet werden.

Weiters soll an dieser Stelle noch darauf hingewiesen werden, dass ein stochastischer Warteprozess, bei dem die Elemente N voneinander unabhängig sind auch als Markow- Prozess bezeichnet wird. Liegen dem Wartesystem diskrete Zeitpunkte t zugrunde spricht man auch von einer Markowschen Kette (vgl. [Neu02], S.668, [HL04], S.732). Diese sind vor allem für Zeitreihenanalysen an stochastischen Modellen interessant bei denen das zukünftige Verhalten des Modells vom vergangenen Verhalten unabhängig ist. Markov-Ketten werden oft wie in Abb. 4.6 dargestellt. Dabei kann zu diskreten Zeitpunkten zwischen den Kettengliedern gewechselt werden. Der Wechsel ist unabhängig vom vorherigen Verhalten.

4.4.2 Das geschlossene Warteschlangennetzwerk

Eines der klassischen Warteschlangenmodelle ist das geschlossenen Warteschlangennetzwerk (**CQN- Modell**, Closed Queuing Network Modell). Dieses Modell arbeitet mit einer konstanten Anzahl von Aufträgen im Netzwerk. Ein Warteschlangennetzwerk besteht dabei aus mehreren Abfertigungseinheiten, die von einem Element durchlaufen werden. Jede Abfertigungseinheit kann wiederum aus mehreren Abfertigungsstationen bestehen und jede Station hat ihre eigene Warteschlange. Dieses System wird dann auch mathematisch sehr komplex. Zur Vereinfachung werden meist unendliche Warteräume, die FIFO- Regel, exponentielle Verteilung der Bearbeitungszeiten sowie gemeinsame Warteschlangen für alternative Betriebsmittel verwendet. CQN-Modelle sind von großer Bedeutung für die Analyse von FFS.

Ein Warteschlangennetzwerk mit N Abfertigungseinheiten und n_i parallelen (identischen) Abfertigungsstationen in den Einheiten i ($i = 1, \dots, N$) wird **Jackson- Netzwerk** genannt (vgl. [Neu02], S.690). Eine Einführung in CQN- Modellierung und die Anwendbarkeit auf FFS findet sich in Tempelmeier [TK93].

Auf Grund der hohen Komplexität werden Warteschlangennetzwerke oft mittels Simulationprogrammen gelöst. Dabei wird das Netzwerk mit Simulationssoftware nachgebildet und durchgerechnet.

4.4.3 Anwendungen für Warteschlangennetzwerke

Colledani, Matta und Tolio ([CMT06], S.77ff) bilden ein Modell zur Effizienzanalyse eines komplexen Fertigungssystem mit zwei verschiedenen Produkten. In einem Fertigungssystem wird die Produktion mittels Markovketten dargestellt. Das Modell berücksichtigt weiters Ausfallswahrscheinlichkeiten der Maschinen und geht von beschränkten Pufferspeichern aus. Der Lösungsalgorithmus wird mit den Ergebnissen einer Simulationsmodellierung verglichen und zeigt dabei annähernd idente Ergebnisse.

Diese Methode ist für automatisierte Montagelinien geeignet und kann auf beliebig viele Produktvarianten erweitert werden.

Für parallele Arbeitsstationen mit gemeinsamer Warteschlange und stochastischen Bearbeitungszeiten wurde von Vuuren, Adan und Resing- Sassen ([VAR06]) ein Algorithmus entwickelt. Dieser lässt sich rascher anwenden als die Erstellung eines Simulationsmodells und kommt auch zu ähnlichen

Ergebnissen.

Bei allen Algorithmen wird jedoch von maschinenorientierter Montage ausgegangen und nicht berücksichtigt, dass die Monteure zwischen den Arbeitsstationen wechseln können.

4.5 Simulation

Die Simulation gehört zum Teilgebiet der experimentellen Mathematik. Bei Simulationen werden vereinfachte Abbilder der Realität nachgebaut. Diese Modelle werden in hinreichender Wiederholung durchgespielt um damit die Auswirkungen verschiedener Einstellungen zu erkennen und Rückschlüsse auf die Realität zu ziehen. Simulationsmodelle dienen deshalb auch nicht primär der Optimierung. Einige Simulationsanwendungen bieten mittlerweile aber schon Optimierungsmöglichkeiten an.

Die mehrmalige Wiederholung ist dabei von entscheidender Bedeutung bei stochastischen Modellen da auch bei gleichen Entscheidungen verschiedenen Ergebnissen erzielt werden können. Deshalb können die Ergebnisse auch nur mittels statistischer Analyse interpretiert werden. Simulationen liefern deshalb im Gegensatz zu exakten Verfahren keine optimalen Lösungen.

Bei einem deterministischen Modell mit bekannten Einflussgrößen die nicht dem Zufall unterliegen sollte man daher auch immer exakte Verfahren wie die lineare Programmierung als Alternative in Betracht ziehen.

4.5.1 Anwendungsgebiete

Simulationen bieten sich bei komplexen, umfangreichen Systemen an die u.U. auch Zufallsgrößen beinhalten können. Aber auch umfangreiche deterministische Modelle mit einer Vielzahl von Variablen sind mittels exakter Methoden oftmals nicht mehr darstellbar. Gerade dann wird auf die Simulation zurückgegriffen. Hauptzielrichtungen der Simulation sind (vgl. [KG95], S. 59):

- Erforschung komplexer Systeme
- Demonstration und Veranschaulichung komplexer Prozesse
- Entscheidungshilfe beim Systemdesign und bei der Systemführung

In der betrieblichen Praxis ist die Simulation ein Instrument für:

- Entscheidungsunterstützung
- Planung
- Organisationsgestaltung
- Kontrolle
- Ausbildung und Training

Die ersten drei Punkte beziehen sich auf die Errichtung neuer Systeme, Anlagen oder Gebäude, Umsetzung neuer Strategien oder auch Änderungen in der Ablauforganisation. Die Kontrolle beinhaltet dabei oft die Machbarkeitsprüfung anstehender Projekte sowie einen anschließenden Vergleich zwischen Realität und Simulation um Fehler zu entdecken und Verbesserungen umzusetzen. Die Simulation als Schulungs- und Trainingsinstrument ist vor allem bei Unternehmensplanspielen und der Pilotenausbildung mit Flugsimulatoren im Einsatz. Simulationsanwendungen in verschiedenster Form sind mittlerweile auch häufig aus Computerspielen bekannt und Gegenstand wissenschaftlicher Artikel (vgl. [Swa05]).

Simulationsmodelle haben also in der Praxis ein breites Anwendungsspektrum. Einige Beispiele finden sich in [HL04], S. 942ff. Es gibt eine Vielzahl von Programmen und Umsetzungsmöglichkeiten für Simulationen. Gerade deshalb sollte der jeweilige Anwendungsfall zuvor genau festgelegt und beschrieben werden. Eine falsche Wahl des Simulationsmodells und damit eine unpassende Umsetzung führen zu unbrauchbaren Ergebnissen. Die Korrektur falscher Modelle bzw. Neuerstellung ist zeit- und kostenintensiv und kann durch vorausschauende Planung vermieden werden.

4.5.2 Einteilung und Klassifizierung von Simulationsmodellen

Simulationsmodelle dienen dem experimentellen Durchspielen von Entscheidungssituationen (...). Man unterscheidet bezüglich des Zeitbezuges zwischen Modellen der statischen und der dynamischen, bezüglich der Datenqualität zwischen solchen der deterministischen und der stochastischen Simulation. [EI93a], S. 1886

Für die richtige Anwendung der Simulationstechnik sind vor allem folgende Kenntnisse erforderlich ([KG95], S.65):

- **Kenntnisse über die Simulation** allgemein. (Was kann die Simulation leisten? Welche Voraussetzungen müssen erfüllt sein? Welche Grenzen hat die Simulation?)
- **Kenntnisse über die verfügbaren Ressourcen** (Softwaretools, Hardware, Fachleute).
- **Kenntnisse über das zu modellierende System** (z.B. Produktionsabläufe, Transportsystem, Fertigungssteuerung usw.).
- **Kenntnisse über alternative Lösungsmöglichkeiten** (z.B. Tabellenkalkulation, Lineare Optimierung, Warteschlangentheorie, statistische Methoden, Kombinatorik, Wahrscheinlichkeitsrechnung usw.).

Wenn nun die Simulation als geeignetes Instrument ausgewählt wird so kann man Simulationsmodelle auch noch bezüglich anderer Kriterien definieren (vgl. [Prab, KG95, Kel02])

- Realitätstreue (abstrakt oder konkret)
- Entstehungsart (natürlich oder künstlich)
- Verhältnis zur Umgebung (geschlossen oder offen)
- Zeitabhängigkeit (statisch oder dynamisch)
- Zeitverteilung (kontinuierlich oder diskret)
- Funktionstyp (linear oder nicht linear)
- Bestimmtheitsgrad (deterministisch oder stochastisch)
- Freiheitsgrade (starr oder flexibel)
- Komplexität des Modells (einfach oder komplex)
- Modellart (physisch oder mathematisch/logisch)
- Art der Simulationssprache (diskret/ interaktive, graphische Systeme/ anwendungsspezifische Systeme)

An Hand von Kriterien ist es möglich, alle Modelle zu klassifizieren. Der Verband deutscher Ingenieure (VDI) bietet dazu die in Abb. 4.7 dargestellte Klassifizierungsmethode an [VDI00].

Bevor mit der Erstellung eines Simulationsmodells begonnen wird sollte überlegt werden, wie das Modell zu klassifizieren ist. Damit kann eine geeignete Simulationssprache (siehe Kapitel 4.6.1) ausgewählt werden. Sich bereits

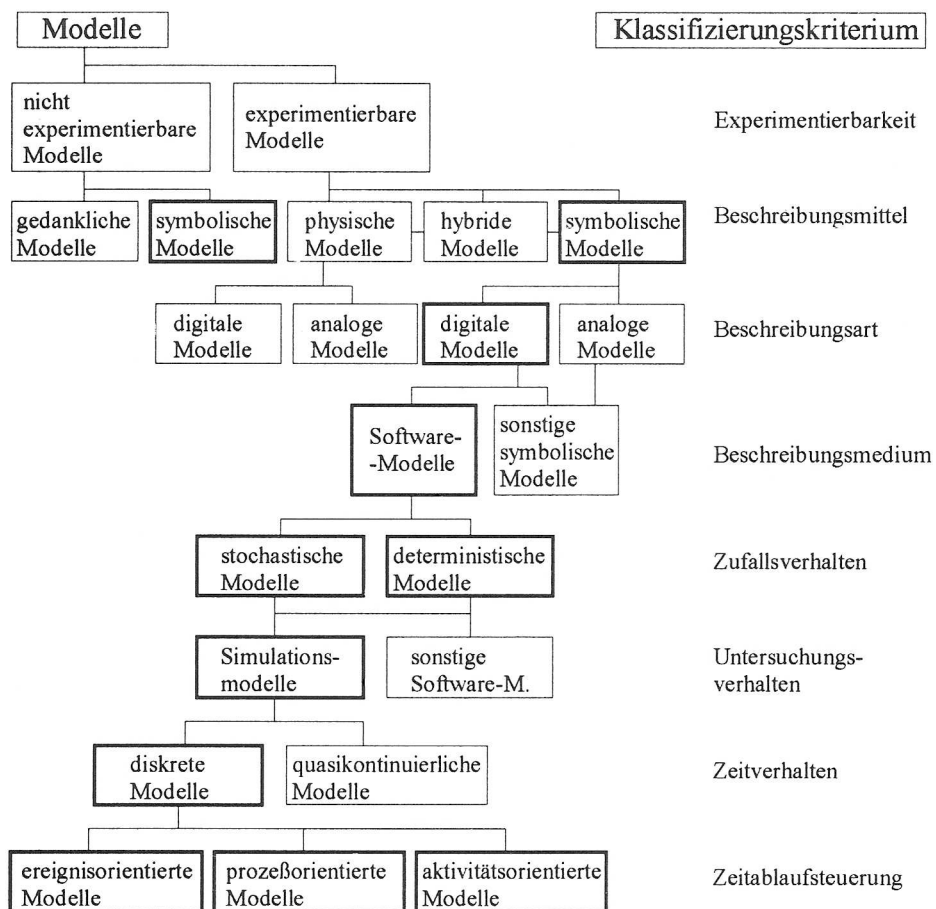


Abbildung 4.7: Klassifizierung von Simulationsmodellen ([VDI00])

in der Vorphase mit der Struktur und dem Aufbau des Modells auseinanderzusetzen hilft auch beim anschließenden Erheben und Auswerten der Daten und der Umsetzung des Modells.

4.5.3 Vor- und Nachteile der Simulation

Die Simulationstechnik hat auf Grund ihres breiten Anwendungsspektrums und ihrer Möglichkeiten eine Reihe von Vorteilen. Auch auf Grund der derzeit verfügbaren Simulationsprogramme und der Rechenleistung herkömmlicher PC's sowie des Wissens auf dem Gebiet der Simulation kann man eine Fülle von Vorteilen aufzählen (vgl. [KG95], S.67, [Tem91], S.2):

Die Entscheidungsqualität und -genauigkeit steigt mit Hilfe der Simulation. Es können beliebige Situationen mit verschiedenen Einstellungen durchgespielt werden. Das ist in vergleichsweise kurzer Zeit möglich, auch wenn das Simulationsmodell Wochen, Monate oder sogar Jahre dauernde Prozesse berechnet. Damit können Fehlentscheidungen vermieden und Kosten gespart werden.

Die Kosten für Simulationsstudien sind im Vergleich zur Investitionssumme bzw. den Plankosten des realen Prozesses vergleichsweise niedrig. Sie betragen weniger als 1% der Investitionssumme bzw. 10% der Plankosten für die reale Umsetzung.

Die Simulation bietet die Möglichkeit, verschiedene Lösungsvarianten zu untersuchen und Sensitivitätsanalysen durchzuführen. Es kann mit verschiedenen (stochastischen) Einflussgrößen experimentiert werden. Damit kann man feststellen, wie und ob sich diese Faktoren auf das Modell auswirken.

An Hand der beschriebenen Anwendungen der Simulation muss man sich auch intensiv mit dem zu modellierenden Problem auseinandersetzen. Diese Tatsache erfordert eine genaue Recherche sowie Zusammenarbeit und Koordination mit allen Beteiligten. Hier muss man sich auch aus verschiedenen Perspektiven dem Problem nähern und auch die Konsequenzen für alle Beteiligten bedenken. Denk- und Modellierungsfehler werden aufgezeigt und die Qualität der Lösung gesteigert.

Die Simulation deckt wie in Kapitel 4.5.1 beschrieben ein breites Anwendungsgebiet ab. Die Simulation ist einfach einsetzbar und kann bei komplexen Systemen rascher zu einer Lösung führen als dies bei analytischen Verfahren

möglich wäre.

Weiters gibt es eine Reihe von Anwendungsfällen, die nur mit Experimenten und nicht durch Versuche am realen Prozess gelöst werden können (vgl. [Kel02], S.5). Die Durchführung wäre in der Praxis nicht möglich, zu teuer oder gefährlich:

- Verschiedene Layouts zur Maschinenaufstellung in einer Fabrik wenn diese noch nicht gebaut ist.
- In existierenden Fabriken und Montagehallen kann das Experimentieren mit der Maschinenaufstellung sehr teuer werden. Besonders große Maschinen oder Kräne können nicht einfach umgestellt werden. Dazu sind oft Umbauarbeiten an der Halle und am Fundament notwendig.
- Es ist schwer möglich, eine erhöhte Auslastung oder Überlastung eines Systems, die auf Grund besonderer Umstände eintritt, gewollt herbeizuführen. Beispielsweise können die Auswirkungen der Schließung einer Bankfiliale und der Kundenansturm auf umliegende Banken in der Praxis nicht getestet werden.
- Auf einem Flughafen neue Check-in Verfahren auszuprobieren kann zu unvorhergesehenen Problemen führen oder auch dazu, dass Passagiere ihren Flug versäumen. Das gilt für alle realen Prozesse. Ein Eingriff in eine bestehende und funktionierende Prozesskette kann zu unvorhergesehenen Problemen an anderen Stellen führen.
- Auch mit der Notaufnahme und den Abläufen in einem Krankenhaus zu experimentieren kann auf keinen Fall zugelassen werden. Hier kann ein Experiment mit dem System sogar lebensgefährlich werden.

Die Simulation ist also vielseitig einsetzbar, kostengünstig und liefert brauchbare Ergebnisse in rascher Zeit. Trotz allem gibt es aber auch Nachteile und Gefahren die man bedenken sollte (vgl. [KG95], S.70, [Tem91], S.2ff): Der Aufbau des Modells erfordert Zeit für die intensive Recherche der Problemstellung, Datenerfassung und Datenvorbereitung. Besonders in dieser frühen Phase sollte sehr genau gearbeitet werden um im späteren Verlauf Modellierungsfehler zu vermeiden. Diese Phase erfordert wesentlich mehr Zeit als die Modellierung selbst und darf nicht unterschätzt werden.

Dabei zeigt sich auch gleich das nächste Problem. Viele der Daten sind nicht in der Genauigkeit vorhanden, die für eine exakte Simulation notwendig wäre. Die Datenbeschaffung kann vor allem für spezielle Modelle schwie-

rig werden, besonders wenn die Betriebsdatenerfassung (BDE) nicht gewartet und aktuell gehalten wird. Bei Modellen, die zukünftige Entscheidungen unterstützen sollen sind Daten überhaupt nicht verfügbar oder müssen geschätzt werden. Andere Daten sind nicht richtig abgelegt und die Suche damit erschwert.

In solchen Fällen müssen Abstriche bei der Genauigkeit des Modells gemacht werden und der Grad der Abstraktion nimmt zu. Sehr abstrakte Modelle bilden nicht mehr die Realität ab oder können auf Grund der Abstraktion die gewählte Problemstellung nicht mehr lösen. Gerade deshalb ist die Verifizierung und Validierung der Modelle ein sehr wichtiger Arbeitsschritt. Der Vertrauensbereich der Aussage muss dabei definiert und überprüft werden.

Der Einfluss von zu vielen stochastischen Größen kann bei der Simulation auch zu stochastischen Ergebnissen führen. Der einmalige Durchlauf einer Simulation darf daher auch nur als einmaliges Experiment aufgefasst werden. Ein Simulationslauf ist nur eine mögliche Ausprägung eines stochastischen Prozesses. Für eine statistische Signifikanz sind daher eine Reihe von Simulationsläufen notwendig. Selbst dann kann das Ergebnis nicht immer das gewünschte statistische Signifikanzniveau erreichen.

Obwohl Simulationssoftware mittlerweile auf Standardrechnern lauffähig ist so sind sie doch Spezialanwendungen die nur von geschulten und erfahrenen Modellierern bedient werden können. Und trotz allem kann die Simulation keine optimale Lösung gewährleisten. Das Ergebnis ist abhängig von den Eingabewerten und der Erfahrung des Benutzers sowie der Interpretation der Daten. Verfahren zur Optimierung von Simulationsexperimenten werden derzeit noch nicht von allen Programmen unterstützt.

Bei nicht sachkundigen Benutzern kann der Einsatz von Simulationsmodellen auf Grund der Fülle von Informationen, die generiert werden den falschen Anschein hoher Glaubwürdigkeit erwecken. Gerade die intuitive und einfache Bedienung sowie die Darstellung mittels graphischer Animation kann den Eindruck vermitteln, eine perfekte Lösung gefunden zu haben die auf jeden Fall richtig ist.

Die Grenzen der Simulation sind vor allem bei sehr großen Systemen, einer Vielzahl von Parametern mit einer starken Korrelation und stochastischen Einflüssen rasch erreicht. Bei unsicheren Eingabedaten kann die Simulation keine verwendbaren Ergebnisse liefern. Simulationen rechnen sich auf

Grund des Zeit- und Kostenaufwandes erst bei längerfristigen Projekten.

4.5.4 Schritte der Simulationsmodellierung

Die Simulationsmodellierung orientiert sich im wesentlichen an der generellen Vorgehensweise der Modellierung in Abschnitt 4.1. Einige Punkte müssen jedoch besonders beachtet werden. Eine gute Anleitung dazu bietet Kosturiak (vgl. [KG95], Kap.6). Die Hauptschritte bei der Erstellung eines Simulationsmodells bestehen aus:

- Situationsanalyse und Zielsetzung
- Datenerhebung und -aufbereitung
- Modellbildung
- Verifizierung und Validierung
- Simulationsexperimente
- Ergebnisanalyse und -auswertung
- Umsetzung im realen System

Die Situationsanalyse und Zielsetzung bei der Simulation muss entsprechend genau erfolgen. Oftmals müssen für die spätere Modellierung Vereinfachungen am Modell vorgenommen werden. Je genauer die Situation beschrieben wird desto genauer kann auch das spätere Modell erstellt werden. Auch die Ziele müssen genau definiert sein. Durch Vereinfachung und Abstraktion am Modell werden sonst möglicherweise wesentliche Kriterien für die Zielerreichung abstrahiert, gestrichen oder verändert. Auch die Unterscheidung zwischen dem Hauptziel sowie den konkret heruntergebrochenen Teilzielen zur Erreichung des Hauptzieles ist wichtig. Gerade diese Teilziele zu identifizieren ist entscheidend für den Erfolg der Simulation. Die Abhängigkeiten zwischen den Zielen zu erkennen und am Modell umzusetzen ist ein wesentlicher Schritt einer guten Modellierung. Die Kunst der Modellierung besteht darin, gegensätzliche Teilziele zu optimieren so dass das Hauptziel am besten erreicht wird.

Der Schritt der Datenerhebung und -aufbereitung schliesst an die Situationsanalyse und Zieldefinition an. Die gewünschten Daten für die Simulation zu sammeln kann schwierig sein, speziell wenn diese von verschiedenen Quellen zusammengetragen werden müssen. Oftmals verleiten die Möglichkeiten

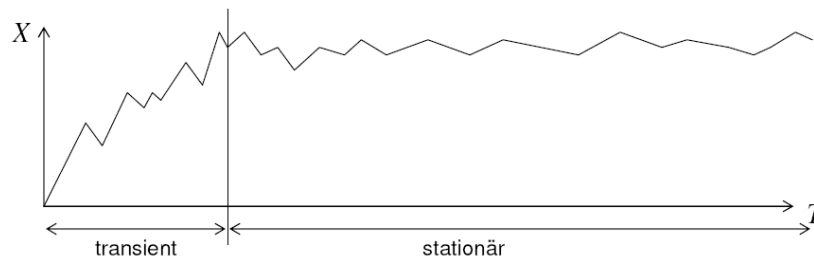


Abbildung 4.8: Anlaufphase eines Simulationsmodells ([Prab], S.21)

der Simulation auch dazu, zu viele Daten zu erheben und in das Modell aufzunehmen. Damit wird das Modell nicht zwangsläufig genauer. Die Daten müssen den Modellanforderungen genügen und zur Zielerreichung beitragen (siehe Abschnitt 4.1).

Die Modellbildung stellt den eigentlichen Kern der Modellierung dar. Dabei ist anfangs zu entscheiden, welche Elemente, Strukturen und Einflüsse des realen Systems überhaupt im Modell dargestellt werden müssen. Hierbei ist die durchgeführte Situationsanalyse hilfreich. Für die Modellierung selbst bieten sich dann zwei Möglichkeiten an:

- Top-down- Modellierung: Vom Allgemeinen zum Speziellen
- Bottom-up- Modellierung: Vom Speziellen zum Allgemeinen

Bei der Top- down- Modellierung wird das Gesamtsystem zuerst grob dargestellt und dann verfeinert bis die gewünschte Genauigkeit erreicht ist. Beim Bottom-up Ansatz wird zuerst ein Teil bzw. mehrere Teile des Gesamtsystems modelliert um diese später zum Gesamtsystem zusammenzusetzen.

Die Verifizierung und Validierung eines Simulationsmodells sind wie bei allen OR- Modellen entscheidende Schritte. Werden diese Schritte nicht ordentlich durchgeführt so leidet darunter die Güte und Genauigkeit des Modells. Viele der modernen Simulationsprogramme bieten Hilfe bei der Verifizierung und melden wenn Fehler entdeckt werden. Die Validierung bleibt jedoch immer noch die Aufgabe des Modellierers.

Ist das Modell nun ausreichend korrekt so werden die Simulationsexperimente durchgeführt. Das Simulationsmodell muss hinreichend oft durchlaufen werden um aussagekräftige und brauchbare Ergebnisse für die folgende Auswertung zu erhalten. Um einen stabilen Systemzustand zu erreichen ist es von maßgeblicher Bedeutung, ob die Simulation eine Anlaufphase benötigt (siehe Abb. 4.8). Entsprechend muss die Vorlaufzeit für die Si-

mulation gewählt werden bevor mit der Datenaufzeichnung begonnen wird. Während der Anlaufphase können die Variablenwerte stark schwanken und damit die statistische Auswertung verfälschen. Objekte, die in ein leeres Modell eintreten haben beispielsweise keine Warteschlange in der sie sich einreihen und warten müssen. Die Warte- und Durchlaufzeit ist damit wesentlich kürzer als das in der Realität der Fall ist. Ein anderes Beispiel ist die Simulation von Verkehrs- oder Transportproblemen. Beim Anlaufen des Modells befinden sich keinerlei Fahrzeuge auf den Straßen und alle Waren können sofort auf leere Fahrzeuge geladen und ohne Verzögerung transportiert werden. Das Modell muss deshalb solange einlaufen, bis der Normalbetrieb erreicht wird. Dies ist erkennbar, wenn die Variablen und Parameter nicht mehr zu stark schwanken. Erst dann kann mit der Datenaufzeichnung begonnen werden.

Als letzten Schritt der Simulationsmodellierung werden die Daten ausgewertet. Für die Datenanalyse werden im Regelfall statistische Methoden angewandt. Die Interpretation der ausgewerteten Ergebnisse erfordert Erfahrung mit dem realen System. Eine falsche Interpretation der Daten und falsche Schlussfolgerungen für das reale System können deshalb auch ein bislang richtiges Simulationsmodell und korrekte Daten als fehlerhaft dastehen lassen. Gerade die Interpretation entscheidet über Erfolg und Misserfolg der Simulation.

Anschließend an die Ergebnisauswertung und Umsetzung kann eine nochmalige Simulation erfolgen. Hier können die Parameter identisch gehalten werden um zu prüfen, wie exakt das Modell die Wirklichkeit vorausgesagt hat. Ebenso kann nun auch die neue Realsituation modelliert werden um weitere Verbesserungen zu erkennen. Damit wäre dann ein vollständig geschlossener Regelkreis wie in Abb. 4.9 realisiert.

Zuerst wird beginnend von einer Ausgangssituation, dem realen System, durch Abstraktion und Vereinfachung das Modell geplant und erstellt. Am Modell werden dann die Experimente durchgeführt. Die Ergebnisse der Experimente werden analysiert und durch den Anwender interpretiert. Auf Grund der Interpretation kann wiederum eine Umsetzung am realen System durchgeführt werden. Dieses neue und überarbeitete reales System kann nochmals analysiert, modelliert und optimiert werden.

Auch dieses Praxisbeispiel orientiert sich an den genannten Schritten und ist in den folgenden Kapiteln entsprechend aufgebaut. Als Handlungsanleitung zur Erstellung eines Simulationsmodelles kann die Prozessdarstellung

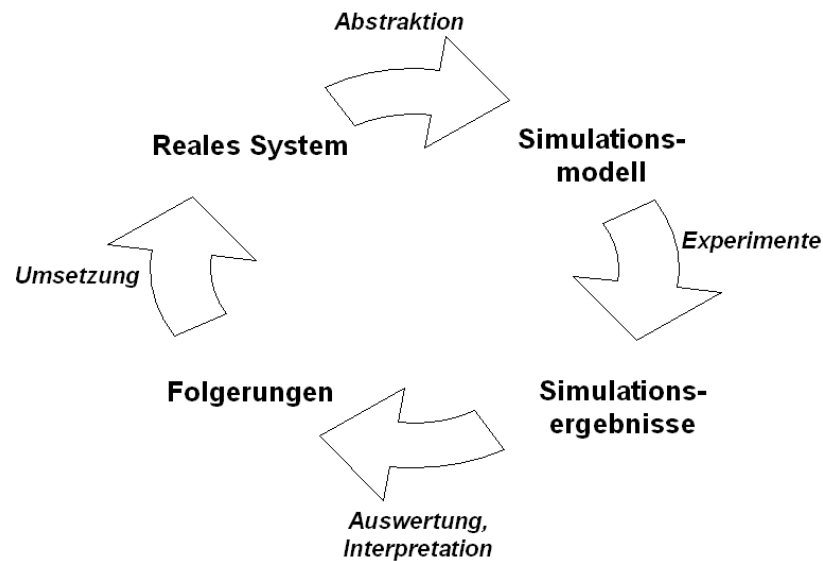


Abbildung 4.9: Phasen einer Simulation

in Abb. 4.10 dienen.

4.6 Simulationssoftware

Eine sorgfältige Auswahl des Simulationssystems in Hinblick auf Kapazität, Abbildungsmöglichkeiten, Benutzerfreundlichkeit und Flexibilität kann die Qualität der Simulation und die Planungskosten erheblich beeinflussen. Die speziellen, anwendungsorientierten Simulationssysteme erfordern meistens keine Programmierkenntnisse und ermöglichen eine relativ schnelle Modellerstellung.
[KG95], S.66

Am Markt werden mittlerweile viele verschiedene Simulationsprogramme angeboten. Diese sind auch auf handelsüblichen PC's lauffähig und können Simulationen in kurzer Zeit durchführen. Damit wird die Simulation von Prozessen auch für KMU's leistbar und ist nicht mehr den Großbetrieben vorenthalten.

Die meisten Simulationssprachen sind der Gruppe der diskreten Sprachen zuzuordnen. Dabei wird zu bestimmten Zeitpunkten ein Ereignis ausgelöst, das Modell neu berechnet und der Systemzustand bestimmt. Sie bietet sich vor allem für stochastische Modelle an da sie den Zufall bzw. die Wahr-

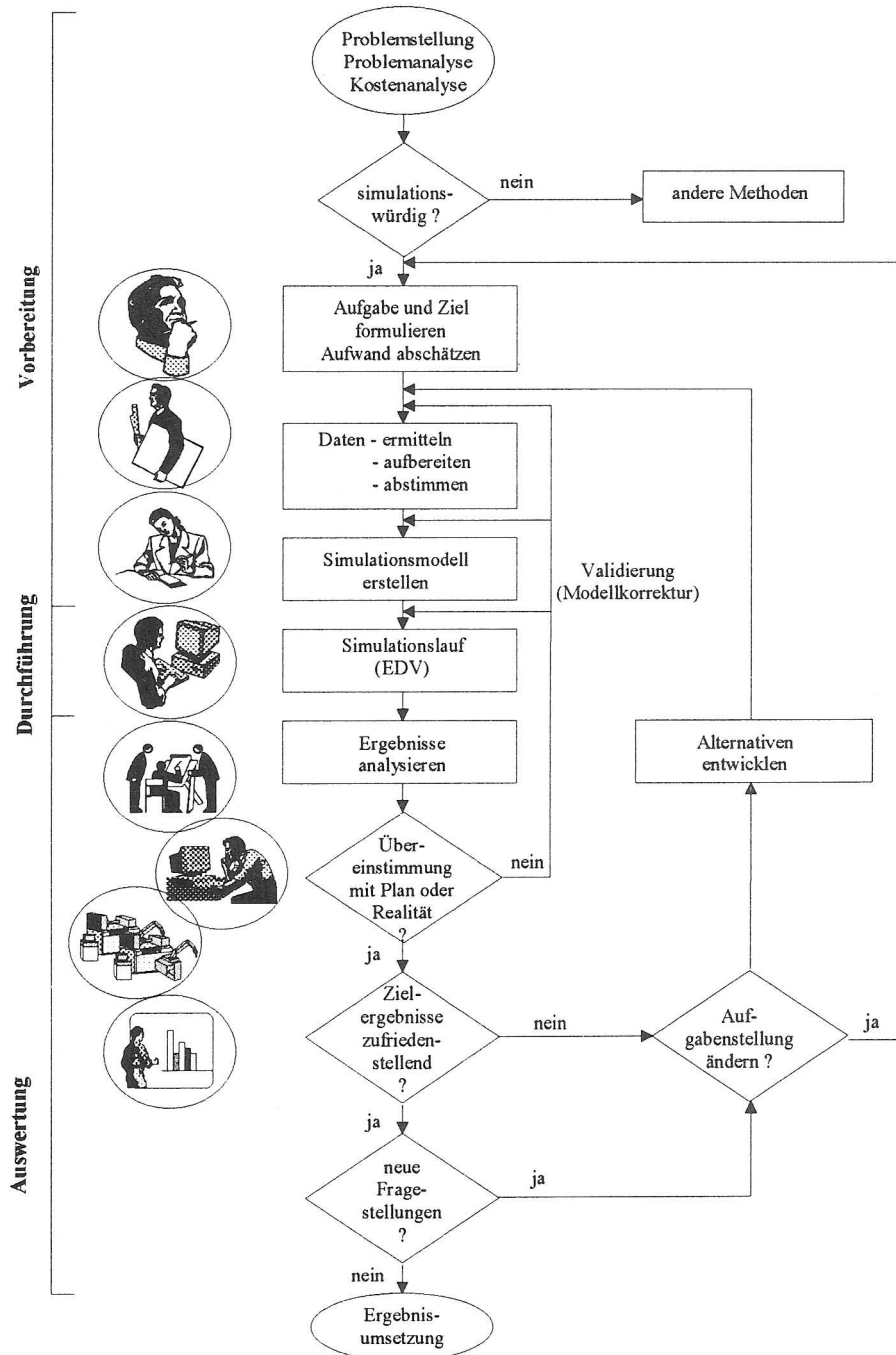


Abbildung 4.10: Ablauf eines Simulationsprozesses ([VDI00])

scheinlichkeit berücksichtigt. Die Programme kann man in folgende Gruppen einteilen (vgl. [Praa]):

- diskrete Simulationssprachen
 - prozessorientiert
 - transaktionsorientiert

- interaktive, graphische Modellierungs- und Simulationssysteme
 - transaktionsorientiert
 - komponentenorientiert
 - Zustandsdiagramme
 - Petri Netze
 - Event Graphs und Activity Cycles

- anwendungsspezifische Modellierungs- und Simulationssysteme
 - interaktive, graphische Modelierung mit anwendungsspezifischen Bausteinen
 - primär in der Fertigungstechnik und Logistik

Man unterscheidet also im wesentlichen zwischen Simulationssprachen und Simulationssystemen. Simulationssprachen bauen direkt auf Programmiersprachen auf und stellen die "Programmiersprache" für die Simulationssysteme. Die Simulationssysteme sind die übliche Software mit der Anwender Simulationsmodelle erstellen und Simulationen durchführen. Auf die Ebene der Simulationssprachen greifen Anwender nur zurück um selbst Module zu programmieren wenn die Simulationssysteme keine fertigen Lösungen anbieten können.

Für die kontinuierliche Simulation gibt es nur wenige Standardsprachen. Eine der bekannten Standardsprachen ist CSMP (Continuous System Modeling Program; IBM). Bei der kontinuierlichen Simulation werden die Abläufe stetig abgebildet. Dabei wird in sehr kurzen Zeitabständen das komplette Modell ständig neu berechnet. Somit entsteht der Eindruck eines kontinuierlichen Systems. Kontinuierliche Simulationsmodelle enthalten eine oder mehrere Differentialgleichungen zur Abbildung des Zusammenhanges zwischen dem Zeitfortschritt und der Änderung der Zustandsvariablen. Die Differentialgleichungen entziehen sich auf Grund ihrer Komplexität allerdings einer analytischen Behandlung (vgl. [Dom07], S.227)

Preislich sind Simulationsprogramme mittlerweile auch für KMU´s leistbar geworden. Die Kosten/ Nutzenrelation wird von Kosturiak an Hand von erfolgreichen Praxisprojekten präsentiert ([KG95], S.107). Die Kosten für die Simulation betragen dabei maximal 13% der in Folge eingesparten Summen. Das ist nicht für alle Simulationen anwendbar, zeigt aber das große Potential an Einsparungen auf, welches mit Hilfe von Optimierungsverfahren wie der Simulation ausgeschöpft werden kann.

4.6.1 Simulationssprachen

Basisprogrammiersprachen, die in die Gruppe der diskreten Sprachen fallen sind GPSS, SIMULA, SIMAN oder Modelica. GPSS (General Purpose Simulation System) ist eine Simulationssprache die primär für die Simulation komplexer Warteschlangensysteme konzipiert ist. SIMULA (SIMulation LAnguage) ist weniger stark problemorientiert und damit flexibler als GPSS. SIMULA basiert auf der Programmiersprache ALGOL. SIMAN (SIMulation ANalysis language) ist ähnlich wie GPSS primär zur Simulation komplexer Warteschlangensysteme gedacht.

SIMAN (SIMulation ANalysis language) zählt zu den diskreten, prozessorientierten Simulationssprachen. SIMAN wurde 1983 von C.D. Pegden entwickelt und basiert auf der Programmiersprache FORTRAN. Es ist eine der Basissprachen, die auch hinter "high-level simulators" steht, beispielsweise ist Arena auf SIMAN- Modulen aufgebaut. Bei der Erstellung eines Modells trennt SIMAN zwischen:

- Modelldefinition und
- Definition des experimentellen Rahmens

In der Modelldefinition wird dabei die Ablaufstruktur des Systems abgebildet, dh. alle durchführenden Prozesse werden beschrieben. Der experimentelle Rahmen erfasst all jene Parameter, welche sich während des Versuchsablaufes ändern können und stochastisch vorliegen. (vgl. [Tem91], S.14ff)

SIMAN erlaubt allerdings auch eine kontinuierliche Simulation, ist damit also eine hybride Programmiersprache. Der Fokus liegt jedoch eindeutig auf der diskreten Simulation.

4.6.2 Simulationsanwendungen

Zwei der bekannteren Simulationsprogramme sind Arena und AnyLogic. Diese fallen in die Kategorie der diskreten, transaktionsorientierten Simulationsmodelle. Sie gehören auch zur Gruppe der "high-level simulators" da sie eine

graphische, intuitiv bedienbare Benutzeroberfläche bieten. Witness, ProModel, Flexsim oder SIMFACTORY sind Beispiele für anwendungsspezifische Simulationssprachen. Einen Überblick über derzeit am Markt befindliche Simulationsanwendungen bietet Swain ([Swa07]). Er veröffentlicht im OR/MS Journal alle zwei Jahre eine aktualisierte Liste der Simulationsanwendungen am Markt die auch online verfügbar ist.¹

AnyLogic ist eine Simulationssoftware die diskrete Simulation, agentbasierte Modellierungstechniken sowie Netzwerkmodellierung ermöglicht. AnyLogic arbeitet mit der Programmiersprache Java und ist objektorientiert aufgebaut. Die Funktionalitäten und die Objektbibliothek von AnyLogic ermöglichen die Modellierung und Simulation verschiedenster Bereiche wie Produktions- und Logistikprozesse, Geschäftsprozesse, Konsumenten- und Patientenverhalten wie auch Umweltprobleme. AnyLogic stellt bereits vorgefertigte Objekten und Entitäten für die Modellierung zur Verfügung. Diese Objekte können direkt in das Modell übernommen werden. Zusätzlich können selbst eigene Objekte erstellt und ergänzt werden. Durch eine relativ offene Java- Umgebung kann das Modell auch mittels Codeprogrammierung erweitert werden. Damit können eigene Methoden verwendet werden. Für den Simulationslauf selbst können dann verschiedene Modelleinstellungen gespeichert werden. Die Auswertung kann direkt in AnyLogic mittels eingebauter Analysetools erfolgen. Es ist auch möglich, die Ergebnisse in andere Programme wie z.B. MS Excel zu übernehmen und auszuwerten. AnyLogic bietet auch eine Reihe von Beispielmotellen für viele verschiedene Anwendungsfälle an die im Programmpaket integriert sind.²

¹siehe <http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html>

²Beschreibung tlw. entnommen von: http://www.simoptimize.ch/software/software_anylogic.html, 01.07.2008

Kapitel 5

Modellformulierung

Aus den Fertigungsstückzahlen in Tab. 2.2 lässt sich eine Verteilung für die 10 verschiedenen Auftragsvarianten ermitteln. An Hand dieser Verteilung sollen die einzelnen Varianten als Objekte zufällig in die Montage gelangen.

Für die erste Auslegung soll eine einzelne Fertigungsstraße modelliert werden. Diese Fertigungsstraße soll alle Varianten bearbeiten können, wobei für Sondermontageschritte (z.B. bei der ABRA) einzelnen Objekte aus der Straße entnommen und anschließend wieder eingefügt werden.

Für die Dauer der einzelnen Arbeitsschritte stehen die Ist- Zeiten der derzeitigen Einzelplatzmontage zur Verfügung. An Hand dieser lässt sich die jeweilige Verteilung für die Dauer der einzelnen Schritte errechnen.

Das gesamte Modell wird auf Basis einer Jahresplanung für die Fertigung erstellt. Die Kapazität der Stationen und Mitarbeiter ist dabei den üblichen Jahresschwankungen auf Grund von Urlaub etc. unterworfen, ebenso wie die Nachfrage nach den unterschiedlichen Fahrzeugen. Über das Jahr verteilt kann es zu Maschinen- und Personalausfällen auf Grund von Wartung, Reparatur, Arbeitsunfällen etc. kommen. Auch diese Ereignisse sollen berücksichtigt werden.

Für die Ablaufplanung des Modells werden folgende Annahmen getroffen. Diese entsprechen nicht der kurzfristigen auftragsorientierten Ablaufplanung wie von Petersen vorgegeben, orientieren sich jedoch an seinem Vorschlag. (vgl. [Pet92], S.70):

1. Das Produktionsprogramm wird an Hand einer Verteilung stochastisch ermittelt

2. Die Bearbeitungszeit an den Arbeitsstationen ist mittels Dreiecksverteilung vorgegeben
3. Jeder Bearbeitungsauftrag ist eine unteilbare Einheit
4. Jede begonnene Operation wird ohne Unterbrechung zu Ende geführt
5. Störungsbedingte Unterbrechungen in den Arbeitsstationen werden nicht berücksichtigt
6. Begrenzte Kapazität der Produktionsfaktoren (Arbeiteranzahl), unbegrenzte Pufferspeicher
7. Die Zeiteinheit im Modell sind Minuten

5.1 Modellannahmen

Für dieses Modell wird angenommen, dass die verschiedenen Auftragsvarianten aus Tab. 2.1 normalverteilt in die Fertigung gelangen. Auf Grund von Zufallszahlen und Verteilungen wird mit Hilfe der Look-up Tabellen Tab. 5.1 und 5.2 ein Fertigungsauftrag entsprechend der Verteilung der Auftragsvarianten aus Tab. 2.2 erzeugt. Das bildet die Wirklichkeit nur beschränkt ab, da Großaufträge einzelner Kunden mit mehreren identischen Fahrzeugen in der Fertigung nicht berücksichtigt werden. Andererseits kommt es auf Grund von Terminverschiebungen auch zu Änderungen im Montageablauf die auf solche Großaufträge keine Rücksicht nehmen.

Argument	Wert	Beschreibung
0	1	4% der Fahrzeuge sind Rotopress- ABRA
0.04	2	32% der Fahrzeuge sind Rotopress
0.36	3	32% der Fahrzeuge sind Variopress
0.68	4	32% der Fahrzeuge sind Variopress SKD

Tabelle 5.1: Lookup table "AuswahlFahrzeugtyp"

Argument	Wert	Beschreibung
0	1	37% haben eine Classic- Verkleidung
0.38	2	63% haben eine Styling- Verkleidung

Tabelle 5.2: Lookup table "AuswahlVerkleidung" für Rotopress

Die Vor- und Hauptfertigung, bei der sich Großaufträge in Form von Serienfertigung auswirken, wird in diesem Modell nicht berücksichtigt. Es wird vielmehr angenommen, dass für sämtliche Fertigungsaufträge die benötigten Bauteile bereits in ausreichender Menge zur Verfügung stehen.

Die verschiedenen Montagezeiten je Arbeitsstation bei den unterschiedlichen Fahrzeugen werden ebenfalls berücksichtigt. Die entsprechenden Montagezeiten werden angepasst und entsprechend passenden Verteilungen auch stochastisch berücksichtigt. Die verwendete Zeiteinheit im Modell sind Minuten. Weiters wird davon ausgegangen, dass eine Woche fünf Arbeitstage und 38,5 Arbeitsstunden hat. Ein Arbeitstag entspricht damit 472 Minuten. Die Verwendung der Einheit Minuten für das Modell hat den Vorteil, die Zeitvorgaben aus den Arbeitsplänen direkt übernehmen zu können.

Für die zu erstellende Simulation wird eine asynchrone Fließfertigung angenommen (siehe Abschnitt 3.1.4). Im Gegensatz zur klassischen Fließfertigung gibt es keine mechanische Einrichtung um die Fahrzeuge weiterzubewegen. Die Fahrzeuge bewegen sich selbst zur nächsten Bearbeitungsstation, sobald sie mit der vorherigen Bearbeitung fertig sind. Daraus ergibt sich die asynchrone Bewegung zwischen den Arbeitsstationen. Damit sind auch Puffer zwischen den Arbeitsstationen notwendig. Für die erste Auslegung des Modells werden diese Puffer mit jeweils 100 Fahrzeugen angenommen. Damit soll dann das Modell validiert werden um die Lauffähigkeit zu überprüfen. In einem zweiten Schritt wird die Puffergröße dann auf ein realistisches Maß minimiert.

Die Arbeiterentitäten sind beweglich und können je nach Bedarf zwischen den Arbeitsstationen wechseln. Einen ähnlichen Ansatz mit Werkzeugen, die zwischen FFS weitergegeben werden können verfolgt Chen ([CC08]). Arbeiterentitäten führen ihre Bearbeitung in einer Arbeitsstation immer komplett aus. Der Ausfall wird dadurch berücksichtigt, dass Entitäten nach erfolgter Arbeitsausführung im Objekt "Arbeitersource" zurückgehalten werden.

5.2 Mathematische Grundlagen

Die Generierung neuer Fertigungsaufträge, die Bearbeitungszeiten in den Stationen sowie die Ausfallszeiten der Arbeiterentitäten werden durch Wahrscheinlichkeitsverteilungen gesteuert. AnyLogic unterstützt eine ganze Reihe von Zufallsverteilungen. Für dieses Modell werden zwei Verteilungen benötigt:

- Normalverteilung für die Auftragsgenerierung sowie die Ausfallszeiten

der Arbeiterentitäten

- Dreiecksverteilung für die Verteilung der Bearbeitungszeiten

In AnyLogic müssen bei der Generierung von wahrscheinlichkeitsverteilten Zufallsgrößen die Art der Verteilung sowie die notwendigen Parameter angegeben werden. AnyLogic erzeugt damit eine der Verteilung entsprechende Zufallszahl. Der Aufruf der entsprechenden Methoden kann direkt in Parametern der aktiven Objekte erfolgen oder über Java- Programmierung vorgenommen werden. Die jeweilige Verteilung mit ihren Parametern wird in den Delay- Objekten im Parameter "delayTime" definiert und aufgerufen. Für die Quelle wird die Verteilung im Parameter "interarrivalTime" definiert.

5.2.1 Die Normalverteilung im Modell

Ankunftsprozesse werden in der Regel anhand einer Exponentialverteilung modelliert. Hier wird das Eintreffen neuer Kundenaufträge (Abb. 5.1) entsprechend einer Normalverteilung mit unterer Schranke generiert (siehe Abb. 5.1). Damit wird nicht das tatsächliche Eintreffen der Kundenaufträge sondern die Abarbeitung des Auftrages bis zur Einlastung ins System durch das technische Büro dargestellt. In der Regel kann vom technischen Büro ein Auftrag bzw. Fahrzeug pro Tag abgearbeitet und für die Fertigung freigegeben werden. Falls Kunden mehrere idente Fahrzeuge bestellen kann die Einlastung auch schneller erfolgen, bei Sonderausstattungen kann die Bearbeitung auch länger dauern. Die Verteilung hat die Form

$$f(\text{Auftrag}) = \max(60, N(462, 180))$$

Der Mittelwert ist $\mu = 462$, die Standardabweichung $\sigma = 180$. Ein neuer Kundenauftrag kommt also im Durchschnitt einmal pro Tag ($7,7h \times 60 = 462min$) in das System mit einer Standardabweichung von drei Stunden. Der minimale Abstand zwischen dem Eintreffen zweier Aufträge beträgt 60 Minuten, dh. eine Stunde. Somit wird der untere Bereich der Normalverteilung abgeschnitten. Wenn keine untere Grenze angegeben wird so wäre es möglich Zeiten < 0 zu erzeugen. Dies würde zu einer Fehlermeldung führen. Die Grenze von 60 Minuten wurde deshalb gewählt, weil es ca. 1 Stunde dauert einen Auftrag in das PPS- System der Firma einzulasten. Die Auftragsart bei Generierung eines neuen Auftrags wird mittels Lookup- Tabelle festgelegt (siehe Abschnitt 5.9).

Die Ausfallszeit der Arbeiter (Abb. 5.2) ist ebenfalls normalverteilt (siehe 5.6.1). Die Verteilung wird monatlich mittels Statechart verändert um die

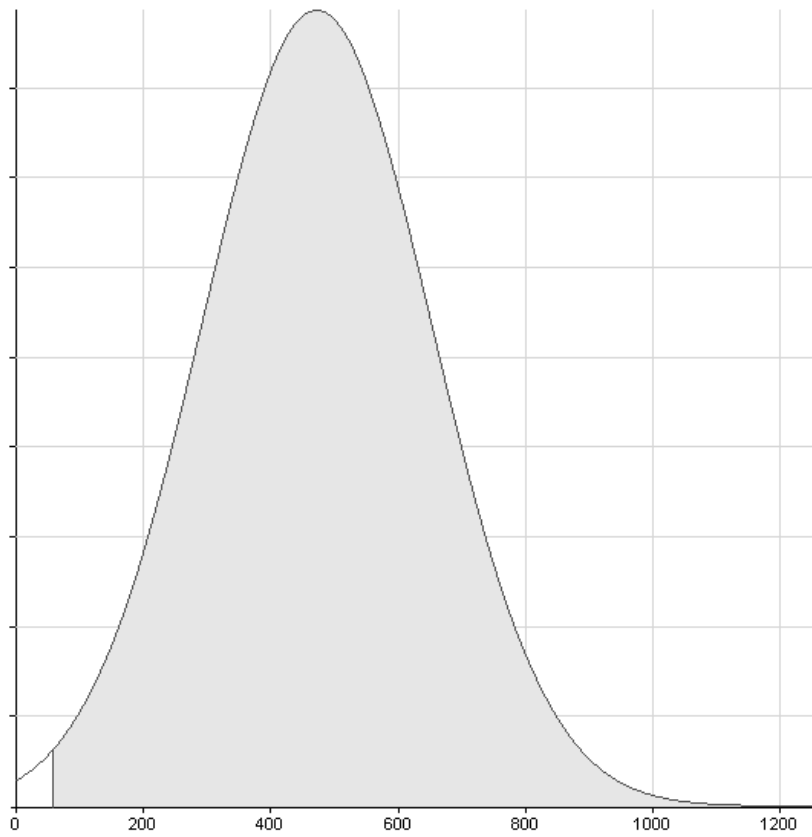


Abbildung 5.1: Dichtefunktion der Auftragsgenerierung

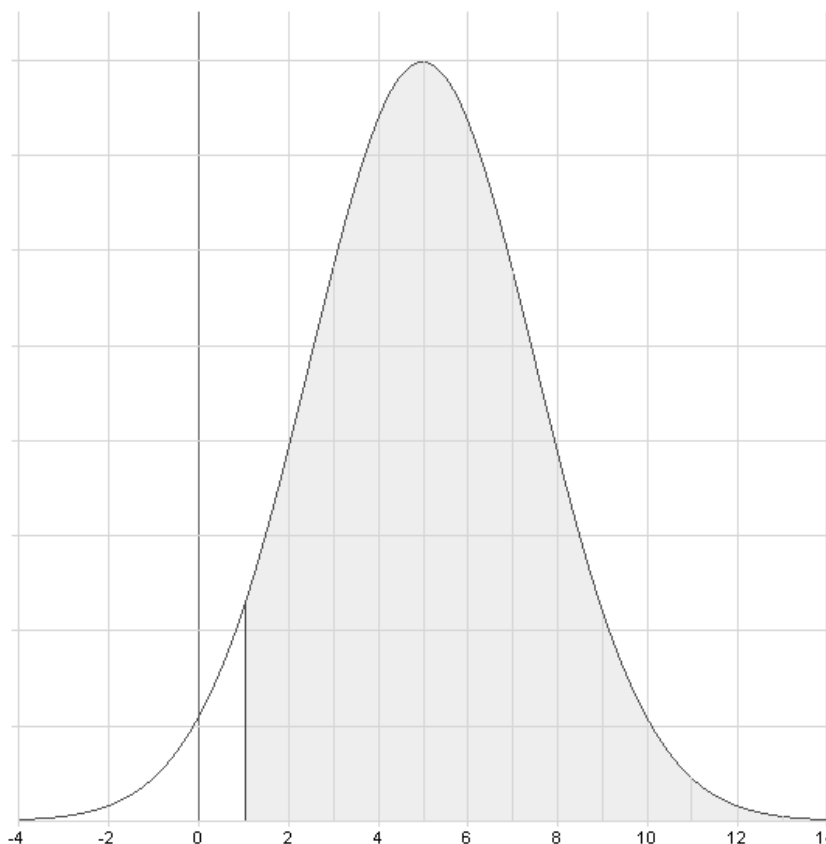


Abbildung 5.2: Dichteverteilung der Ausfallszeiten der Arbeiterentitäten

realen Verhältnisse möglichst genau modellieren zu können (siehe Abschnitt 5.6.1). Die stetige normalverteilte Ausfallszeit entspricht jedoch nicht den tatsächlichen Gegebenheiten. Arbeiter können Urlaub, Krankenstandstage etc. in der Regel nur in ganzen Arbeitstagen konsumieren. Deshalb werden die mittels Normalverteilung ermittelten Ausfallszeiten noch auf ganze Arbeitstage normiert und auf Arbeitsminuten umgerechnet.

5.2.2 Die Dreiecksverteilung im Modell

Die Dreiecksverteilung erfordert den Wert mit der größten Wahrscheinlichkeit m , dem minimalen Wert a und dem maximalen Wert b des Intervalls in dem die Zufallszahlen liegen. Die Dreiecksverteilung verwendet man oftmals bei groben Simulationsberechnungen, wenn die genaue Vorstellung über die Verteilung der Zufallsgröße fehlt. ([KG95], S. 31)

Die Wahrscheinlichkeitsfunktion der Dreiecksverteilung ergibt sich durch:

$$f(x) = \begin{cases} \frac{2(x-a)}{(b-a)(m-a)} , & \text{wenn } a \leq x \leq m \\ \frac{2(b-x)}{(b-a)(b-m)} , & \text{wenn } m \leq x \leq b \end{cases}$$

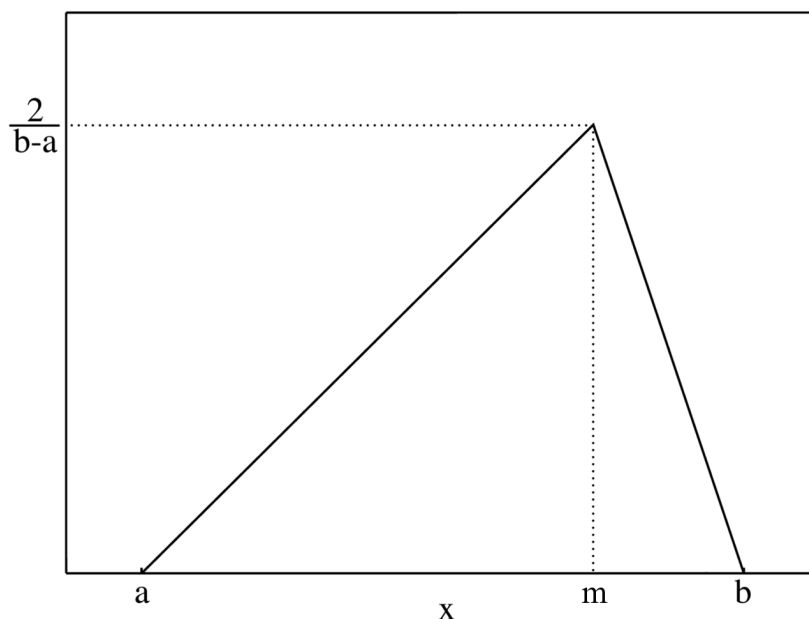


Abbildung 5.3: Dreiecksverteilung

Die Arbeitszeiten für die einzelnen Montageschritte wurden aus den aufgezeichneten tatsächlichen Bearbeitungszeiten ermittelt und als Dreiecksverteilung in das Modell übernommen. Für jeden Arbeitsschritt ist im PPS-System ein Arbeitsplan vorhanden der auch Vorgabezeiten enthält. Für jeden Arbeitsplan werden die IST- Zeiten aufgezeichnet. Die IST- Zeiten unterliegen dabei einer Reihe von Schwankungen. Als eine der Haupteinflussgrößen wurde dabei die Lernkurve angenommen. Diese hätte sich in der Verteilung im Rahmen einer linksschiefen Verteilung auswirken sollen. Zu Beginn wären die Montagezeiten noch länger, würden sich dann jedoch mit zunehmender Produktionserfahrung verringern.

Tatsächlich trat dieser Effekt nicht auf. Zum Einen gibt es kaum noch Lerneffekte da die eingesetzten Arbeiter mit ihrer Tätigkeit vertraut und schon eingearbeitet sind. Andererseits kommt es bei verschiedenen Varianten der

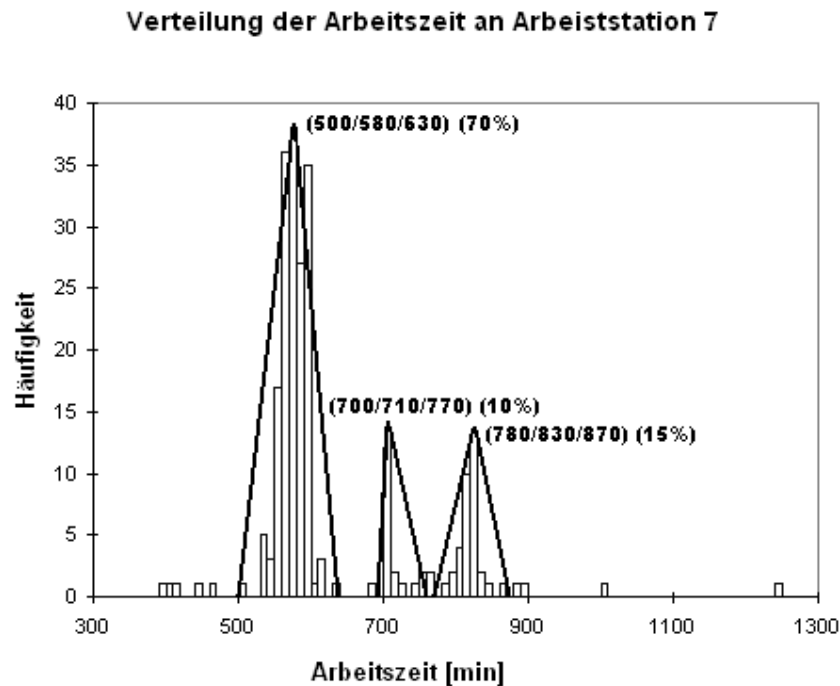


Abbildung 5.4: Dreiecksverteilungen der Bearbeitungszeiten Rotopress an Station 7

Montage zu unterschiedlichen Bearbeitungszeiten die sich in mehreren Maximalwerten ausdrücken. Die tatsächlichen Verteilungen (Beispiel siehe Abb. 5.4) sind also mittels klassischer statistischer Verteilungen nicht modellierbar. Deshalb bietet es sich an, hier zur vereinfachten Dreiecksverteilung zu greifen. Diese kann anschließend über die tatsächlichen Verteilungen gelegt werden.

Bei Verteilungen mit mehreren Spitzen wie in Abb. 5.4 werden zusätzliche Dreiecksverteilungen modelliert. In diesem Simulationsmodell können bis zu drei verschiedene Spitzen je Arbeitsstation und Auftragsvariante definiert werden. Wenn die Verteilung der entsprechenden Arbeitszeit im Modell aufgerufen werden soll wird zuerst zufällig eine der Verteilungen gewählt. Die Auswahl einer der drei Spitzen erfolgt an Hand der Häufigkeitsverteilung, mit der die Dreiecksverteilung in der gesamten Stichprobe vorkommt. Die tatsächliche Arbeitszeit wird dann aus der gewählten Stichprobe bestimmt. Die Auswahl einer Verteilung und Zuweisung der Arbeitszeit zur Arbeitsstation wird in einer Java- Methode programmiert. Diese wird bei jedem Ferti-

gungsauftrag der in einer Arbeitsstation wartet neu aufgerufen. Die exakte Methode ist in Anhang B abgebildet.

5.3 Prioritätsregeln

Prioritätsregeln (Vorrangregeln) sind eine

einfache Form von heuristischen Verfahren für die Reihenfolgeplanung, insb. die Maschinenbelegungsplanung. Sie geben an, welcher Auftrag als nächster an einem Arbeitsträger zu bearbeiten ist, wobei sie eines oder mehrere Merkmale der Aufträge bzw. der Arbeitsträger zur Auswahl des nächsten Auftrages heranziehen, die Prioritätszahlen vergleichen und jene mit dem höchsten Werte wählen. [EI93a]

Bei den Prioritätsregeln hat man in AnyLogic die Wahl zwischen:

- LIFO
- FIFO
- PRIORITY
- RANDOM

Bei allen Warteschlangen von AnyLogic ist standardmässig die FIFO-Regel hinterlegt. Entitäten werden also entsprechend der Reihenfolge ihres Eintreffens aus der Warteschlange entnommen. Für jede Warteschlange kann auch eine der drei anderen Prioritätsregeln eingestellt werden. FIFO führt dazu, dass die zuletzt eingelangte Entität zuerst entnommen wird. Bei RANDOM wird ein zufälliges Element der Warteschlange entnommen. Mit der Einstellung PRIORITY kann über den Parameter "priority", welchen jede Entität besitzt, die Reihenfolge der Einlastung in die Warteschlange gesteuert werden. Das Feld "priority" ist ganzzahlig. Die Entitäten werden entsprechend ihrem "priority"- Wert in absteigender Reihenfolge in die Warteschlange eingereiht.

Mittels Programmierung lassen sich beinahe alle Prioritätsregeln die sonst noch in der Literatur vorkommen über die PRIORITY- Regel in AnyLogic realisieren. Ein Vergleich zu den verschiedenen Prioritätsregeln bzgl. deren Anwendbarkeit bei verschiedenen Zielsetzungen findet sich bei Aspiron ([Asp00], S.268). Zu den Prioritätsregeln siehe auch [Sch02], S.265ff [TK93],

S.219, [Pet92], S.73ff. Im Modell wird die FIFO- Regel für die Kundenaufträge sowie die Arbeiter verwendet. Die Kundenaufträge werden also entsprechend ihrem Einlagen an den Arbeitsstationen bearbeitet. Für die Arbeiter ist die Prioritätsregel nicht relevant, es könnte auch jede andere Regel gewählt werden. Der Arbeitsauftrag wird vom nächsten freien Arbeiter ausgeführt. Für die Arbeitsaufträge wird die PRIORITY- Regel verwendet. Der Prioritätswert wird bei der Erstellung des Arbeitsauftrages im Objekt "Bearbeitung" ermittelt und dem Arbeitsauftrag zugewiesen (siehe auch Abschnitt 5.6.2 sowie für den Programmcode Anhang B). Der Wert wird dabei folgendermaßen bestimmt:

$$priority = L_{QueueAuftrag} + stationsnr$$

Der Prioritätswert addiert sich aus der aktuellen Länge der Warteschlange bei der Arbeitsstation sowie der Stationsnummer. Je länger die Warteschlange, desto höher wird der Prioritätswert. Damit werden also den Arbeitsstationen bei denen mehr Kundenaufträge warten bevorzugt Arbeiter zugeteilt. Weiters fließt auch die Stationsnummer ein. Die Stationen sind entsprechend der Bearbeitungsfolge durchnummeriert. Damit haben bei gleicher Länge der Warteschlange Arbeitsstationen die weiter hinten in der Bearbeitungsfolge stehen einen Vorteil gegenüber den ersten Arbeitsstationen. Während der Verifizierungsphase hat sich gezeigt, dass die Arbeiter immer nur den ersten Stationen zugewiesen wurden, jedoch fast nie den nachfolgenden Stationen. Dadurch waren sehr viele Aufträge im System, es wurden aber keine Aufträge abgeschlossen.

5.4 Datensammlung und Auswertung

AnyLogic unterstützt die Anbindung von Datenbanken um Werte in das Modell übernehmen zu können. Die Anbindung erfolgt über JDBC (Java database connectivity technology) und muss manuell durchgeführt werden. JDBC ermöglicht die direkte Anbindung von MS Access und MS Excel- Dateien, alle anderen Datenbanken können über die ODBC- Schnittstelle (open database connectivity) angebunden werden. Die Datenverwaltung in Datenbanken bietet sich besonders bei einer großen Zahl von Daten an, wenn diese extern geändert werden oder auch für andere Modelle zur Verfügung stehen sollen. Deshalb wurden die ausgewerteten Zeiten je Arbeitsstation in der MS Access-Datenbank gespeichert. Die Daten werden dann mittels der Methoden "zeitenAnfragen" (siehe Anhang B.3 und B.4) in den Objekten Bearbeitung und Lackierarbeiten eingelesen. Die Methode wird für jeden Fertigungsauftrag bei jeder Arbeitsstation durchgeführt.

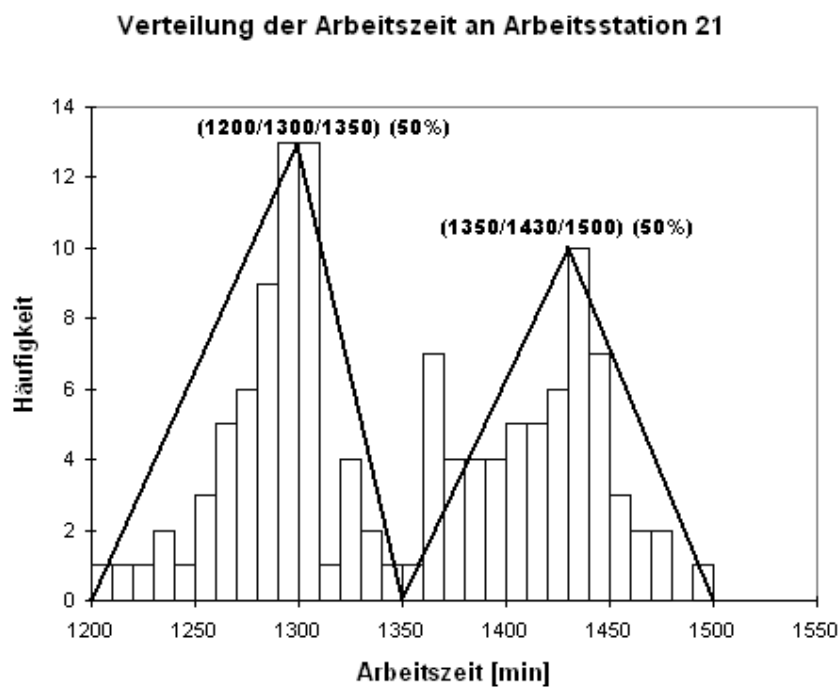


Abbildung 5.5: Dreiecksverteilungen der Bearbeitungszeiten Variopress an Station 21

5.5 Layouterstellung

Für die Simulation wird das Programm AnyLogic verwendet. AnyLogic ist für die Modellierung großer und komplexer Systeme ausgelegt. Jedes neue Modell wird in AnyLogic als Projekt angelegt. Ein Projekt definiert ein komplettes Modell. AnyLogic-Projekte bestehen im wesentlichen aus aktiven Objekten (*active objects*) und Entitäten (*messages*). Aktive Objekte werden für die Modellierung diverser Objekte der realen Welt verwendet. Dazu gehören beispielsweise Arbeitsstationen, Ressourcen oder physische Objekte. Aktive Objekte sind unbeweglich und dienen zur Manipulation und Steuerung der Entitäten. Entitäten sind Instanzen von *message-class* Objekten die sich zwischen den aktiven Objekten entlang von Pfaden bewegen können. Zusätzlich können noch Variablen (*variables*), Parameter (*parameter*), Sequenzdiagramme (*state charts*), Formeln (*enumeration*) sowie Funktionen (*mathematical function* oder *algorithmic function*) und Timer definiert werden um Objekte und Entitäten zu manipulieren. Daten des Simulationslaufs können in *datasets* gespeichert werden. Mittels Animation ist es möglich, 2D- oder 3D-Animationen des Modells zu erstellen.

Da AnyLogic auf der objektorientierten Programmiersprache Java basiert sind alle Objekte aus Klassen aufgebaut. Jedes aktive Objekt ist eine Instanz der *active object class*. Selbes gilt auch für alle Entitäten. Diese stellen Instanzen der *message class* dar. Dieser objektorientierte Ansatz bietet den Vorteil, dass Strukturen leicht aufgebaut und Objekte im Modell öfter verwendet werden können. Ein definiertes aktives Objekt kann daher mehrmals verwendet werden. In den Bibliotheken findet sich eine Reihe von vordefinierten aktiven Objekten welche in eigenen Projekten verwendet werden können. Es können auch eigene aktive Objekte erstellt und in selbst definierten Bibliotheken abgelegt werden. Weiters gilt wie bei allen Java-Klassen, dass Merkmale vererbt werden können. Klassenparameter, Strukturelemente und Methoden der Superklasse treffen auch auf die Sub-Klasse zu.

Auf Grund dieser Vererbung ist AnyLogic hierarchisch aufgebaut. Aktive Objekte können andere aktive Objekte beinhalten. Diese Verschachtelung kann beliebig tief erfolgen. Jedes Projekt in AnyLogic hat bei der Neuanlage ein aktives Objekt (in der Regel "*Main*") als Basisklasse. Alle anderen aktiven Objekte sind Teil dieser Basisklasse oder Teil anderer aktiver Objekte. Durch diese Verschachtelung ist es möglich, Modelle trotzdem übersichtlich darzustellen. Komplexe Abläufe mit vielen Objekten werden in einem eigenen aktiven Objekten dargestellt und erscheinen nur noch als ein Objekt. So sind beispielsweise im Objekt "*Bearbeitung*" die Abläufe einer Arbeitsstation

zusammengefasst. Im "Main"- Objekt gibt es nur noch Instanzen dieses Objekts an Stelle sämtlicher Objekte, die bei einer Arbeitsstation durchlaufen werden.

Alle aktiven Objekte haben Ein- und Ausgänge (*ports*). Diese werden durch Konnektoren (*connector*) miteinander verbunden. Die Verbindungen können fest modelliert, mittels Java- Code implementiert oder dynamisch während des Simulationslaufes verändert werden.

Im Allgemeinen können Einstellungen und Parameter für alle Objekte in AnyLogic direkt definiert oder mittels Javacode festgelegt werden. Beinahe alle Einstellungen können dabei auch variabel und während des Simulationslaufes veränderbar sein. Damit ist es möglich, auch sehr komplexe veränderbare Systeme mit AnyLogic darzustellen.

Für eine detaillierte Einführung in AnyLogic und dem Aufbau aller Objekte siehe 5.5. Dieses Manual ist im Lieferumfang von AnyLogic enthalten.

Der Fertigungsablauf des Modells lässt sich wie in Abb. 5.6 bis Abb. 5.9 als Prozess darstellen. In der Quelle werden die Arbeitsaufträge entsprechend der Dichteverteilung generiert und durchlaufen die einzelnen Arbeitsstationen im Modell. Dabei werden einige Arbeitsstationen nur von bestimmten Auftragsvarianten durchlaufen. Entsprechend der Darstellung werden zu Beginn alle neuen Arbeitsaufträge gesplittet und auf zwei Fertigungsstraßen parallel bearbeitet. In einer der Straßen wird der Auftrag je nach Fahrzeugtyp nochmals in "Rotopress" und "Variopress" unterteilt. Die drei Straßen führen danach wieder zusammen und es erfolgt die weitere Bearbeitung im Modell. Im Anschluss an die gemeinsame Weitergabe erfolgen die weiteren Arbeitsschritte. Bei jedem Prozessschritt ist in der linken oberen Ecke auch die zugehörige Stationsnummer angegeben.

5.6 Aktive Objekte

Instanzen von aktiven Objekten werden aus Objektklassen erstellt. Diese können aus den bestehenden Bibliotheken übernommen oder selbst definiert werden. Alle Objektklassen haben einen Klassennamen, Parameter und eine grafische Darstellung im Strukturdiagramm. Im folgenden werden die selbst definierten Objekte für dieses Modell erklärt. Die Strukturdiagramme sowie Parametereinstellungen sind in Anhang A ersichtlich.

Für das Modell werden folgende Klassen aktiver Objekte (*active object class*) definiert:

- Arbeitersource

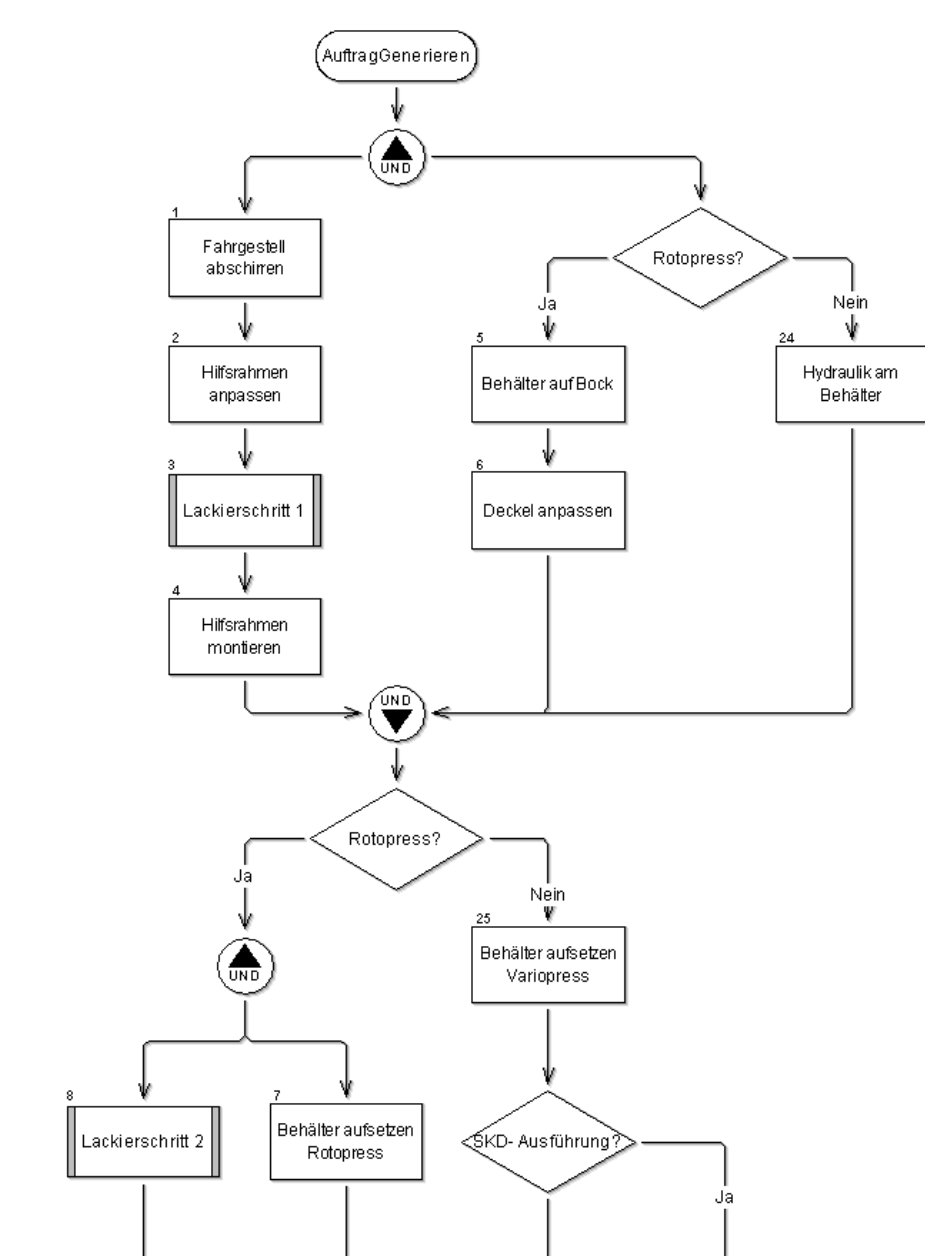


Abbildung 5.6: Der komplette Fertigungsprozess

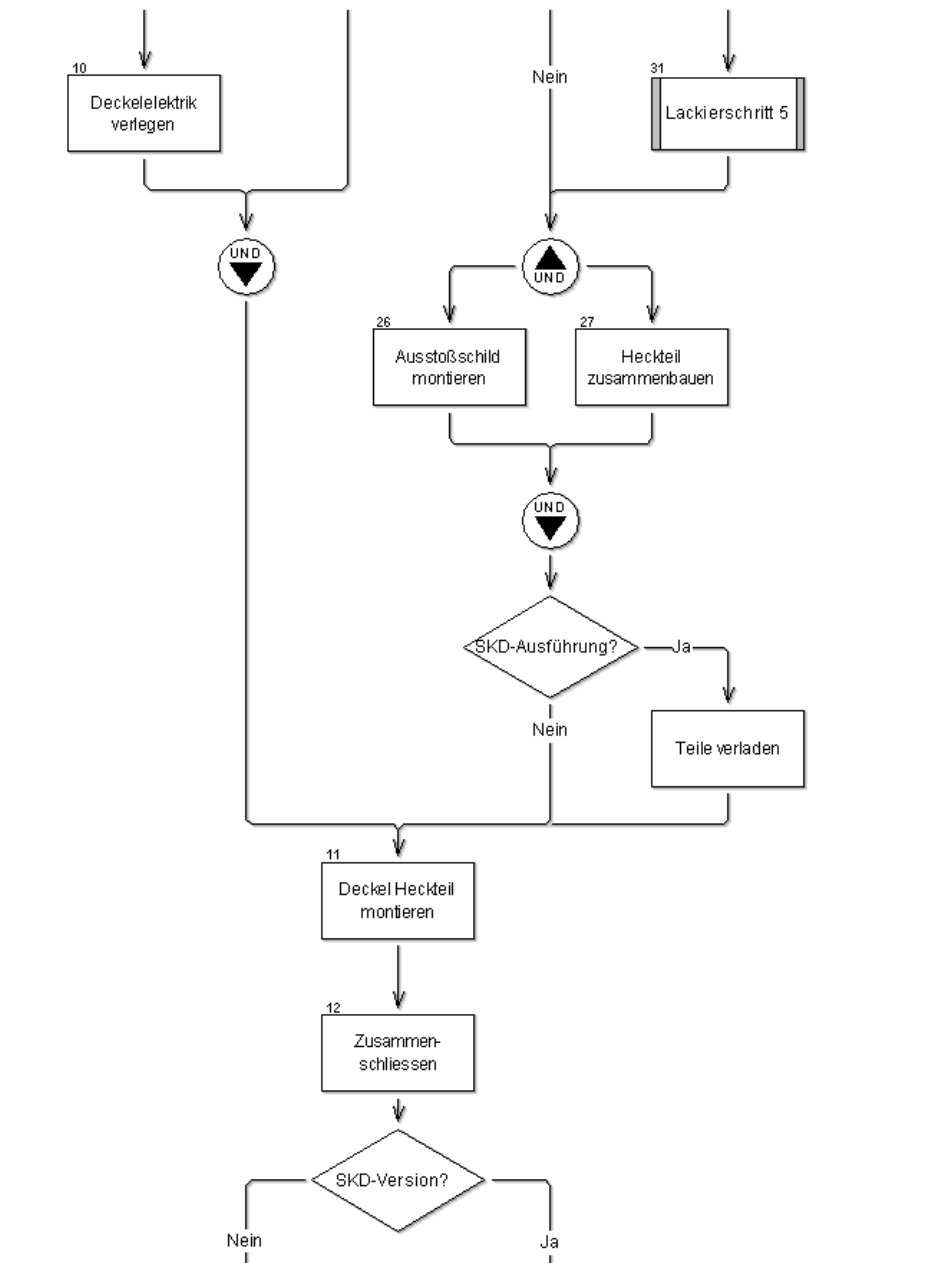


Abbildung 5.7: Der komplette Fertigungsprozess (Forts.)

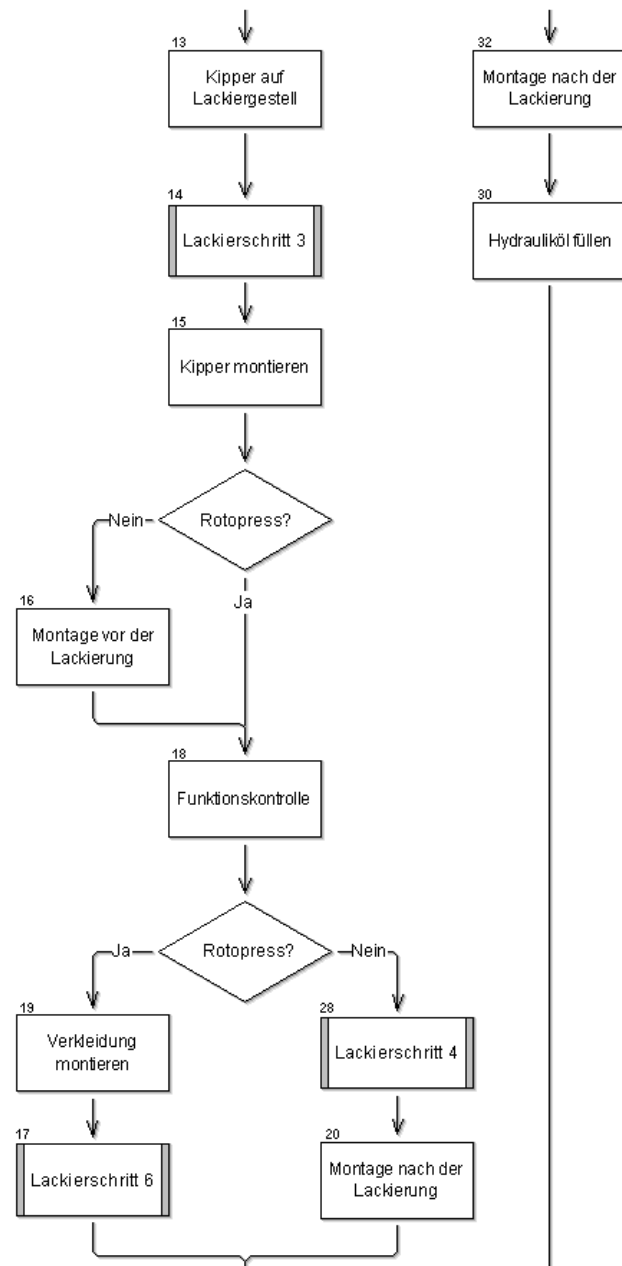


Abbildung 5.8: Der komplette Fertigungsprozess (Forts.)

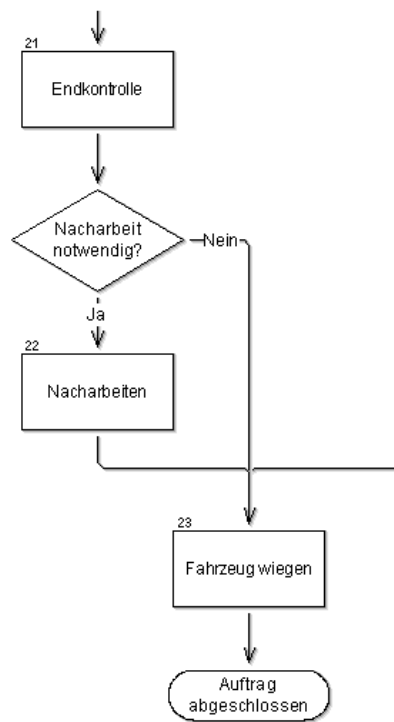


Abbildung 5.9: Der komplette Fertigungsprozess (Forts.)

- Bearbeitung
- Gemeinsame Weitergabe
- Lackierarbeiten

5.6.1 Objekt "Arbeitersource"

Das Objekt "Arbeitersource" dient zur Erzeugung und Steuerung der verschiedenen Arbeiterentitäten (siehe Abschnitt 5.7.3). Dieses Objekt ist für alle Arbeiterentitäten identisch, dh. jede Arbeiterentität wird von einer Instanz des Objekts erzeugt und verwaltet. Der Ablauf innerhalb des Objektes ist in Abb. 5.10 ersichtlich. Beim Setup des Modells werden sofort alle Arbeiterentitäten erzeugt und in die Warteschlange (Pkt. 3) gestellt. In der Warteschlange bleiben die Arbeiterentitäten so lange, bis ein Arbeitsauftrag eintrifft. Die Arbeitsaufträge werden von den Arbeitsstationen erzeugt und fordern eine Arbeiterentität an, die die Bearbeitung durchführt. Je Arbeitsauftrag kann eine Arbeiterentität geholt werden. Die Auftragsentität wird vernichtet und die Arbeiterentitäten verlassen dann die Arbeitersource um die Bearbeitung in der Arbeitsstation durchzuführen. Nach der erfolgten Bearbeitung betreten die Arbeiterentitäten das Objekt wieder. Danach wird abgefragt, ob die Entitäten ausfallen oder sofort wieder in die Warteschlange gereiht werden. Die Ausfallswahrscheinlichkeit sowie die Verteilung der Ausfallsdauer wird monatlich angepasst (siehe unten). Weiters ist die Gesamtzahl von Arbeitern, die gleichzeitig ausfallen dürfen, limitiert. Damit wird verhindert, dass in Simulationsläufen plötzlich überproportional viele Arbeiter nicht anwesend sind. Solche Extremsituationen werden auch in der Realität durch Urlaubssperren oder Ersatz- bzw. Leiharbeiter verhindert.

Das Modell geht davon aus, dass Ausfälle von Arbeitern erst nach der erfolgten Bearbeitung in einer Station passieren können. Arbeitsaufträge in Stationen werden also immer abgeschlossen und nicht unterbrochen. Das ist für eine einzelne Entität, wenn diese als Person interpretiert wird, nicht korrekt. Arbeitsunfälle passieren während der Arbeitszeit und auch Urlaube werden zu den genehmigten Terminen konsumiert. Das kann auch während der Bearbeitung in einer Station passieren. Im Falle, dass eine einzelne Person ausfällt wird ein anderer Arbeiter die begonnene Tätigkeit jedoch mit sehr geringere Verzögerung zu Ende bringen. Der Ausfall von Arbeiterentitäten kann daher als Verkleinerung des Pools an verfügbaren Arbeitern gesehen werden und ist für die Gesamtheit des Modells zulässig.

Für dieses Objekt wird auch noch ein statechart definiert (siehe Anhang A.1.1). Statecharts ermöglichen das Verhalten von aktiven Objekten während

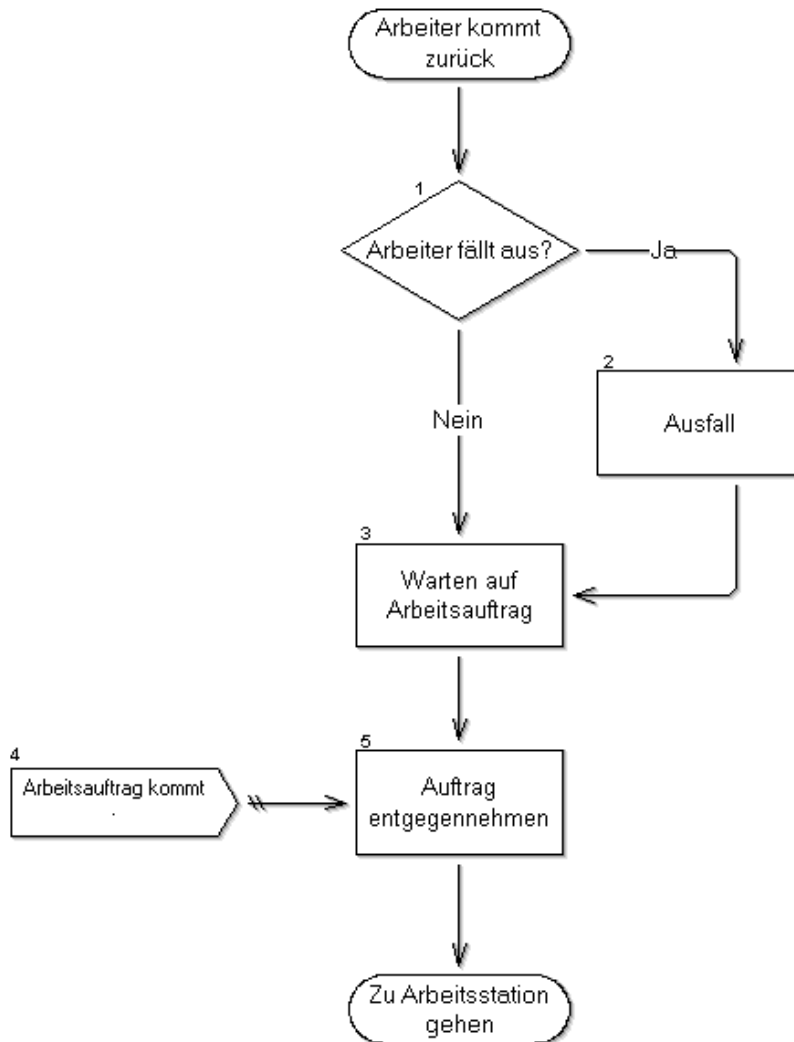


Abbildung 5.10: Ablaufdarstellung für das Objekt "Arbeiter"

des Modelllaufes zu verändern. Die Auslöser für diese Veränderungen können dabei ereignis- oder zeitgesteuert auftreten.

Für die Arbeitersource wird dabei die Wahrscheinlichkeit sowie Dauer für einen Ausfall eines Arbeiters während des Zeitverlaufs geändert. Den größten Ausfallsanteil an produktiver Zeit verursachen Urlaube und Krankenstände. Da die Urlaubs- und Krankenstandswahrscheinlichkeit über das Jahr nicht gleich verteilt ist werden die Wahrscheinlichkeiten monatlich angepasst. Als Basis zur Berechnung der Wahrscheinlichkeiten wurden dabei die Ausfallszeiten des letzten Geschäftsjahres erhoben und ausgewertet. Auch die Dauer eines einzelnen Ausfalls ist nicht konstant. Während der Haupturlaubszeit im Sommer beträgt der Ausfall oft zwei Wochen, im Winter dagegen nur eine Woche. Auch Krankenstände die eine Häufung im Herbst zeigen sind oft nur auf wenige Tage beschränkt.

Die Wahrscheinlichkeiten sowie Dauer der Ausfälle werden im Statechart deshalb monatlich geändert. Die Ausfallswahrscheinlichkeit $P(Urlaub_m)$ wird damit für das Ereignis "Arbeiter fällt aus" (Nr. 1 in Abb. 5.10) sowie die Ausfallsdauer L_m für den Schritt 2 "Ausfall" verändert. Die Ausfallsdauer L_m beträgt mindestens einen Arbeitstag (462 Minuten = 7,7 Std.) und kann um den monatlich geänderten Mittelwert l_i schwanken. Der jeweilige Wert wird mittels Normalverteilung ermittelt die dann auf ganze Arbeitstage normiert wird (siehe Abschnitt 5.2.1). Im Zuge der Validierung wurden mittels eines Ausfallsfaktors die Wahrscheinlichkeit $P(Urlaub_m)$ noch angepasst damit die Ausfallsdauer L_m den realen Werten annähernd entspricht.

$$P(Urlaub_m) = p_m \times \text{Ausfallfaktor}$$

$$L_m = \max(462, N(l_i, l_i/2))$$

$$m \in \{1, 2, \dots, 12\}$$

Dieser Ausfallsfaktor kann je nach Arbeiterart für jedes der drei Instanzen Schweißer, Schlosser und Elektriker separat eingegeben werden. Das ist notwendig da Schweißer viele kurze Arbeitsaufträge bearbeiten, Schlosser hingegen oft längere Aufträge. Da die Ausfallswahrscheinlichkeit nach Beendigung jedes Arbeitsauftrages neu ermittelt wird würden Schweißer überproportional oft ausfallen, Schlosser dagegen zu selten.

5.6.2 Objekt "Bearbeitung"

Dieses Objekt stellt die einzelnen Arbeitsstationen der Montage dar. Der Ablauf innerhalb des Modells ist vereinfacht in Abb. 5.11 dargestellt. Kun-

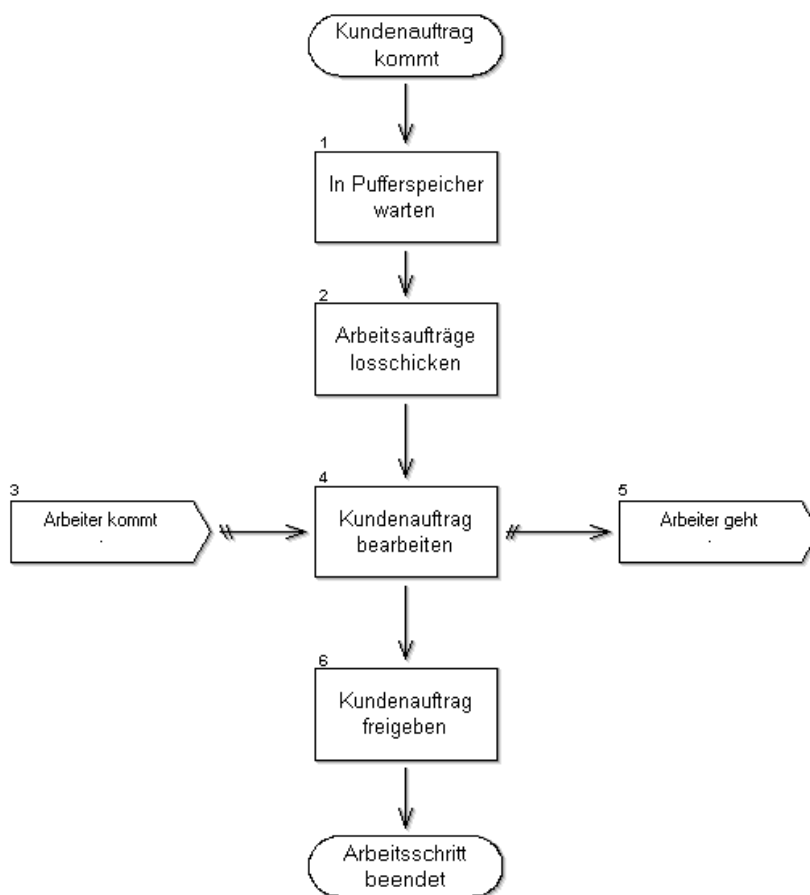


Abbildung 5.11: Ablaufdarstellung für das Objekt "Bearbeitung"

denaufträge betreten das Modell und warten im Pufferspeicher auf die Bearbeitung. Für den jeweils ersten Auftrag in der Warteschlange wird die Methode "ressourcenAnfragen" ausgeführt und die benötigten Arbeiterentitäten aus den Objekten "Arbeitersource" geholt. Die Warteschlangenlänge, dh. die Größe des Pufferspeichers, kann je Arbeitsstation separat eingegeben werden. Damit können auch reale Pufferkapazitäten je Arbeitsplatz modelliert werden.

Jeweils ein Kundenauftrag in der Station wird bearbeitet. Dieser steht im Objekt 4 "Kundenauftrag bearbeiten" der Abb. 5.11. Er bleibt solange in Bearbeitung, bis alle Arbeiterentitäten ihren Bearbeitungsschritt abgeschlossen haben. Die unterschiedlichen Arbeiterentitäten können die Bearbeitung voneinander unabhängig durchführen. Bei Eintreffen eines Arbeiters aus dem Objekt "Arbeitersource" wird die Methode "zeitenAnfragen" ausgeführt um die Bearbeitungszeit der Arbeiterentität für diesen Auftrag zu

ermitteln. Die Arbeiterentität führt den Bearbeitungsschritt dann durch und kehrt wieder in die "Arbeitersource" zurück. Sind alle Bearbeitungsschritte durchgeführt dann wird der in Bearbeitung befindliche Auftrag freigegeben und an die nächste Station übergeben. Im Anschluss werden wieder die nächsten Arbeitsauftrags- Entitäten losgeschickt um die Arbeiterentitäten für den nächsten Kundenauftrag anzufordern.

Die einzelnen Arbeiterentitäten sind in Wirklichkeit während der Bearbeitung nicht komplett unabhängig voneinander tätig. Diese Annahme wurde zur Vereinfachung getroffen da das Modell sonst noch wesentlich komplexer geworden wäre. In der Praxis arbeiten Schlosser sehr lange an einem Arbeitsschritt während die anderen Arbeiterklassen nur kurze Bearbeitungsschritte dazwischen durchführen. Das ist mit der Annahme der Unabhängigkeit möglich da auch hier die Bearbeitung zumeist parallel erfolgt. Lediglich der genaue Startzeitpunkt der gemeinsamen Bearbeitung ist nicht definiert.

5.6.3 Objekt "Lackierarbeiten"

Dieses Objekt bildet im Gegensatz zum Objekt "Bearbeitung" nicht eine einzelne Lackierstation sondern die gesamte Lackiererei ab. Die zu lackierenden Teile bzw. Fahrzeuge werden aus dem Montageprozess entnommen, in die Lackierstation überstellt und lackiert und anschließend wieder zurückgebracht. Die Lackiererei bekommt dabei die Teile aus sechs unterschiedlichen Phasen des Montageablaufes zur Bearbeitung und muss diese in den drei zur Verfügung stehenden Lackierboxen bearbeiten.

Weiters werden auch die benötigten Arbeiterklassen "Stapler" und "Lackierer" nicht mittels dem Objekt "Arbeitersource" erzeugt und verwaltet sondern vom Objekt "Lackierarbeiten" selbst. Da dieses Objekt die gesamte Lackiererei abbildet stellt es ein in sich geschlossenes System dar bei dem nur die Kundenauftragsentitäten das Objekt betreten und verlassen können. AnyLogic bietet für solche Fälle das Standardobjekt "ProcessQ" an. Hier können mehrere idente Arbeitsstationen (Lackierboxen) einfach durch Änderung des Parameterwertes "delayCapacity" dargestellt werden. Jede ProcessQ kann fix mit einem Ressourcenpool verbunden werden aus dem die Arbeiterentitäten geholt werden.

Ein Kundenauftrag wartet in einem Pufferspeicher der Montage auf die Abholung durch den Stapler. Die Stapler müssen sowohl Teile zur Bearbeitung abholen als auch wieder zurückbringen. Die jeweiligen Transportzeiten sind dabei als fixe Größen angenommen. Der Rücktransport von Teilen aus

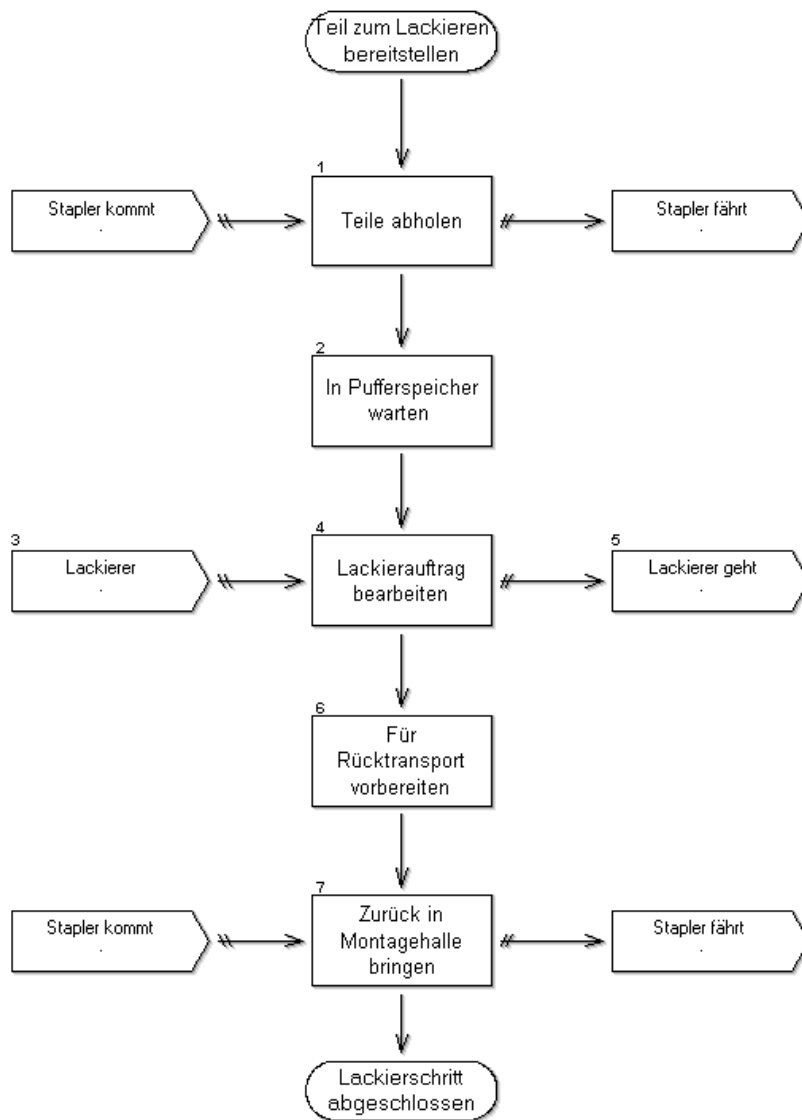


Abbildung 5.12: Ablaufdarstellung für das Objekt "Lackierarbeiten"

der Lackiererei in die Montagehalle hat dabei Vorrang gegenüber der Abholung der Teile. Der Lagerplatz der Lackiererei für Fertigteile ist sehr beschränkt. Weiters würden Teile vom Modell sonst nie zurückgebracht sondern nur abgeholt werden. Die definierten Transportzeiten gehen aber davon aus, dass es keine Leerfahrten gibt.

Nach dem Abholen der Teile warten diese in dem Pufferspeicher der Lackierboxen auf die Bearbeitung. In den drei Lackierboxen kann jeweils ein Kundenauftrag bearbeitet werden und vor jeder Box kann ein Kundenauftrag warten. Der Ressourcenbedarf je Kundenauftrag muss in der Lackiererei nicht erhoben werden da je Box bzw. Auftrag genau ein Lackierer benötigt wird und diese Größe nicht veränderbar ist. Deshalb wird bei jedem Kundenauftrag nur die Bearbeitungszeit mittels der Methode "zeitenAnfragen" festgelegt. Nach der erfolgten Lackierarbeit erfolgt der Rücktransport analog zum Abholen der Teile.

Lackierer- und Staplerentitäten fallen auch niemals aus. Das wäre zwar für einzelne Arbeiter bzw. Stapler in der Realität zutreffend. In der Praxis wird jedoch immer sofort ein Ersatz für Stapler bzw. Staplerfahrer gefunden und auch für die Lackierboxen steht immer ein Lackierer zur Verfügung. Diese vereinfachende Annahme konnte deshalb auch für das Modell angewendet werden.

5.6.4 Objekt "Gemeinsame Weitergabe"

Dieses Objekt dient dazu, zusammengehörende Teile aus verschiedenen Fertigungsstraßen wieder zu kombinieren. Wie in Abb. 5.6 bis Abb. 5.9 ersichtlich können sich Fertigungsstraßen nochmals splitten. Dadurch ist es möglich, dass Kundenaufträge auf einer Straße schneller bearbeitet werden als auf der anderen Straße und damit die Reihenfolge des Eintreffens bei der Zusammenführung der beiden Linien unterschiedlich ist. Hier einfach das "combine"- Objekt aus AnyLogic zu verwenden könnte dazu führen, dass verschiedene Fertigungsaufträge miteinander verbunden werden.

In diesem Objekt gibt es daher zwei Eingänge mit separaten Warteschlangen in denen die Kundenaufträge der beiden Linien eintreffen. Sobald zwei Kundenaufträge mit gleicher Auftragsnummer in den Warteschlangen stehen werden diese zu einer gemeinsamen Kundenauftragsentität vereint und verlassen das Objekt durch den Ausgang. Weiters werden die Bearbeitungs- und Wartezeiten für die neue Entität aus den beiden bereits bearbeiteten Kundenauftragsteilen übernommen und addiert.

Das Gegenstück zu diesem Objekt ist das "split"- Objekt welches einen Kundenauftrag auf zwei Fertigungsstraßen aufteilt. Dieses Objekt ist ein Standard- Objekt aus AnyLogic und wird hier nicht näher erklärt.

5.7 Entitäten

Entities Entitäten (*messages*) werden in AnyLogic zu diskreten Zeitpunkten zwischen den aktiven Objekten weitergegeben. *Messages* stellen die beweglichen Objekte eines Modells dar, werden entlang von Konnektoren bewegt und treten durch die Ports in die Objekte ein und wieder heraus. Sie entsprechen den Objekten und Personen der Realität an denen Tätigkeiten ausgeführt werden oder die aktive Objekte bedienen. Während der Interaktion der Objekte mit den Entitäten können zu diskreten Zeitpunkten Aktionen ausgelöst oder Java- Methoden aufgerufen werden. Entitäten betreten aktive Objekte durch Ports. Konnektoren stellen die Pfade dar, auf denen sich die Entitäten bewegen. *Messages* können mit Parametern belegt werden und verschiedene Informationen transportieren. Diese Informationen können auch an aktive Objekte übergeben und während des Simulationsablaufs verändert werden.

Alle Entitäten im Modell stellen Instanzen von *message*- Klassen dar. *Message*- Klassen existieren nicht als Vorlagen sondern müssen immer selbst erstellt werden. Alle *Message*- Klassen haben einen Klassennamen und müssen einer Java- Basisklasse zugeteilt sein. Die Klasse *Object* ist dabei voreingestellt. Für das Modell werden folgende Klassen von Entitäten (*message classes*) benötigt:

- Arbeiter
- Arbeitsauftrag
- Kundenauftrag
- Lackierer
- Stapler

Die Parameterwerte sowie genaue Definition aller Entitäten steht in Anhang A.

5.7.1 Kundenauftrags- Entitäten

Die Entität "Kundenauftrag" stellt die Kundenaufträge dar die im System eintreffen und bearbeitet werden. Die Entitäten werden stochastisch an Hand der Verteilung im Geschäftsjahr 2007/08 (siehe Tab. 2.2) in der Quelle erstellt. Am Ende der Bearbeitung wird die Entität nach dem letzten Fertigungsschritt in der Senke vernichtet. Die Entitäten werden im Fertigungsfluss bei paralleler Bearbeitung gesplittet. Dabei wird eine Kopie der jeweiligen Entität erstellt die dann in der parallelen Fertigungsstraße läuft. Beim Zusammentreffen der beiden Fertigungsstraßen im Objekt "Gemeinsame Weitergabe" wird dann die Kopie vernichtet. Jede Entität zeichnet während der Bearbeitung die Bearbeitungszeiten und Wartezeiten auf. Die Bearbeitungszeit B_i ist die Summe der Bearbeitungszeiten in allen Arbeitsstationen

$$B_i = \sum b_{ai}$$

Analog dazu ist die Wartezeit W_i die Summe der Wartezeiten in den Warteschlangen aller Bearbeitungsstationen

$$W_i = \sum w_{ai}$$

Wegen teilweiser paralleler Arbeitsschritte ist die Summe aus Bearbeitungs- und Wartezeit nicht die Durchlaufzeit. Die Durchlaufzeit wird aus der Differenz zwischen Auftragsgenerierung in der Quelle und Auftragsvernichtung in der Senke berechnet und stellt somit analog zu den Netzplänen den längsten Weg durch die Fertigung dar.

$$T_i = T_{ei} - T_{ai}$$

5.7.2 Arbeitsauftrags- Entitäten

Die Arbeitsauftragsentitäten fordern die Arbeiter für die Arbeitsstationen an. Sie tragen nur die Nummer der Arbeitsstation in der sie generiert wurden. Wenn ein Kundenauftrag in der Warteschlange des Objekts "Bearbeitung" steht werden im Objekt "Bearbeitung" die Arbeitsauftragsentitäten generiert und losgeschickt. Die Arbeitsaufträge gehen dann in die entsprechenden "Arbeitsource"- Instanzen und warten dort auf einen freien Arbeiter. Die Arbeitsaufträge werden dabei entsprechend ihrem Prioritätswert in die Warteschlange eingereiht (siehe Abschnitt 5.3).

Wenn ein Arbeiter frei ist nimmt dieser den nächsten Arbeitsauftrag entgegen und bewegt sich zur entsprechenden Arbeitsstation um die Bearbeitung durchzuführen. Der Arbeitsauftrag wird vernichtet wenn er von einem Arbeiter entgegengenommen wird.

5.7.3 Arbeiter- Entitäten

Die Arbeiterentitäten repräsentieren die verschiedenen Arbeiter im Modell. Jede Art von Arbeitern wird in einem eigenen "Arbeitersource"- Objekt generiert. Das Arbeitersource- Objekt stellt den Warteraum für Arbeiter dar. Sie nehmen Arbeitsaufträge entgegen, gehen zu den Arbeitsstationen um diese zu bearbeiten und kehren wieder in die Arbeitersource zurück. Für alle Entitäten wird die Ausfallszeit sowie die Arbeitszeit aufgezeichnet. Auch die Wartezeit in allen Warteschlangen wird aufgezeichnet. Diese Einzelwerte können je nach Entität stark schwanken und haben für sich genommen keine Aussagekraft. Für das Modell relevant sind die aggregierten Werte die in den Variablen der Arbeitersource- Instanzen zusammengefasst werden.

Die Lackierer- und Staplerentitäten werden direkt im Objekt "Lackierarbeiten" verwaltet. Diese Entitäten verlassen dieses Objekt nicht sondern sind ihren Arbeitsstationen fix zugewiesen (siehe auch Abschnitt 5.6.3 sowie AnhangA). Es werden auch keine Werte für die einzelnen Entitäten aufgezeichnet. Für das Modell wird angenommen, dass die Anzahl der Entitäten konstant ist und es zu keinen Ausfällen kommt. Das stimmt auch mit der Realität überein da bei Ausfällen von Lackierern und Staplerfahrern auf Grund von Urlauben, Krankenständen etc. sofort andere freie Arbeiter oder Leiharbeitskräfte einspringen.

5.8 Variablen

Variablen werden dazu verwendet, Modelldaten im Zeitverlauf darzustellen sowie Ergebnisse zu dokumentieren. Variablen halten den Modellstatus fest und können sich im Zeitverlauf ändern. Parameter auf der anderen Seite sind statisch. Diese verändern sich nicht über den Zeitablauf sondern, wenn überhaupt, nur zu bestimmten Aktionen. Sie steuern das Verhalten von Objekten. Variablen können auch innerhalb von Objektklassen definiert werden und gelten dann nur für Instanzen dieses Objektes sowie allen darunterliegenden Instanzen. Variablen des "Main"- Objektes gelten für alle Objekte im Projekt da diese auf der obersten Ebene liegen. Für das Modell werden folgende Variablen (*dataset*) im Objekt "Arbeitersource" definiert:

- *urlaubszeit*
- *arbeitszeit*
- *wartezeit*

- *urlaubsdauer*
- *urlaubswahrscheinlichkeit*

Die ersten drei Variablen dienen zur Aufzeichnung der kumulierten Ausfallszeiten, Bearbeitungszeiten und Wartezeiten der einzelnen Entitäten. Bei jeder Entität werden diese drei Werte separat aufgezeichnet und daraus die monatliche kumulierte Summe gebildet und in die lokalen Variablen geschrieben. Diese Variablenwerte werden für jedes Monat kumuliert ausgegeben und können dann weiter ausgewertet werden.

Die beiden Variablen *urlaubsdauer* und *urlaubswahrscheinlichkeit* dienen der Steuerung der Ausfallswahrscheinlichkeit und -dauer der Arbeitentitäten. Mittels statechart werden die Werte monatlich angepasst (siehe Abschnitt 5.6.1).

5.9 Weitere Steuerelemente

Für das Modell werden folgende Zähler (*enumeration*) benötigt:

- Warten
- Bearbeiten
- Startzeitpunkt
- Endzeitpunkt

Lookup-Tabellen (*Lookup-table*) werden benötigt um komplexe nicht-lineare Zusammenhänge darzustellen. Lookup-Tabellen werden dann verwendet, wenn eine Werteberechnung mittels einer linearen oder nicht-linearen Funktion nicht mehr möglich ist. In Lookup-Tables werden Werte bzw. Wertebereiche für bestimmte Funktionswerte hinterlegt. Für das Modell werden die *Lookup-Tables* "AuswahlFahrzeugtyp" (Tab. 5.1) und "AuswahlVerkleidung" (Tab. 5.2) benötigt.

5.10 Verifizierung und Validierung

Nach erfolgter Modellerstellung ist es notwendig, die Korrektheit des Modells zu prüfen. Bei der Verifizierung unterstützt AnyLogic den Anwender. AnyLogic generiert Java-Code für alle Klassen und Instanzen im Modell. Mit dem

eingebauten Java- Compiler wird das gesamte Modell übersetzt und dabei der Syntax überprüft. Wenn dabei Fehler auftreten werden diese von AnyLogic ausgegeben. Damit kann die Verifizierung rasch und einfach durchgeführt werden.

AnyLogic ist jedoch nicht in der Lage, die Werte von Parametern, Variablen, etc. auf ihre Richtigkeit zu prüfen. Dieser Schritt zur Prüfung der modellinternen Logik und Konsistenz bleibt dem Anwender selbst überlassen. Um den korrekten Aufruf sowie die Ausführung aller Aktionen prüfen zu können wird das Modell schrittweise validiert. Von einer einfachen und starren Grundeinstellung werden nach und nach alle Funktionen im Modell dazugenommen und der Ablauf überprüft. Die Prüfschritte im Verlauf sind:

- Durchlaufprüfung der Entitäten
- Datenbankanbindung an Access prüfen
- Engpassanalyse und Modellkorrektur
- Prüfung des Statechart für die Arbeiter- Entitäten
- Vollständiger Modelldurchlauf und Validierung der Fertigungszeiten

Zuerst erfolgte eine reine Prüfung, wie sich die Entitäten durch das Modell bewegen. Die Wahrscheinlichkeit für den Ausfall der Arbeiterentitäten wurde auf $p_i = 0$ gesetzt. Die Bearbeitungszeiten wurden dabei in allen Stationen konstant auf $t_i = 5$ gesetzt und die Warteschlangenlänge $q_i = 100$. Die Arbeiteranzahl wurde für die Klassen Schlosser, Schweißer und Elektriker jeweils auf $a_n = 100$ gesetzt. Damit konnte geprüft werden, ob Entitäten im Modell fehlgesteuert werden, verloren gehen oder im Modell hängen bleiben.

Dann wurde die Datenbankanbindung durchgeführt um die Bearbeitungszeiten auftragsspezifisch korrekt einlesen zu können. Hier konnte auch schon abgeschätzt werden, wo die Engpässe des Modells liegen würden. Die Anbindung der Datenbank führte auch zu keiner nennenswerten Verlangsamung des Modells obwohl bei jeder Arbeitsstation und jedem Arbeitsauftrag eine Abfrage durchgeführt werden muss.

Anschließend erfolgte der vollständige Modelldurchlauf mit der Engpassanalyse. Die Warteschlangenlänge wurde dabei noch immer auf $q_i = 100$ belassen. So lange Warteschlangen sind in diesem Fertigungssystem schon alleine auf Grund der Größe der Bauteile und des damit verbundenen enormen Lagerplatzbedarfes gar nicht möglich. Die Minimierung der Schlangenlänge q_i

auf reale Werte erfolgt allerdings erst bei den tatsächlichen Untersuchungen. In diesem Schritt sollte nur die Bearbeitungszeiten in den Stationen ermittelt und mit den bekannten IST- Werten verglichen werden. Dabei zeigte sich, dass auf Grund der bereits erkannten Engstellen nach einer simulierten Laufzeit von ca. 1,2 Jahren die Warteschlangen an den Engstellen voll waren und die Simulation abgebrochen werden musste. Als besonders kritisch wurden dabei die Arbeitsstationen 4, 11 und 27 eingestuft. Das sind drei sehr arbeitsintensive Stationen. Zwei der Stationen sind in parallelen Arbeitslinien. Lange Bearbeitungszeiten halten damit auch die parallele Linie auf. Deshalb wurden die Engstellen jeweils um eine zweite Bearbeitungsstation des selben Typs ergänzt um den Durchfluss zu erhöhen. Durch die Ergänzung jeweils einer zusätzlichen Arbeitsstation konnte die Zeit bis zum Programmüberlauf mehr als verdoppelt werden. Die Laufzeit von 2,5 Jahren bis zum Fehlerabbruch war somit für die zu simulierende Laufzeit von einem Jahr ausreichend.

Nun wurde die Warteschlangenkapazität auf den realistischeren Wert von $q_i = 10$ reduziert, die Arbeiteranzahl blieb auf $a_n = 100$. Durch die kürzeren Warteschlangenkapazitäten wurde nun das Limit rascher erreicht. Dadurch wurden die Engstellen noch klarer erkennbar. In mehreren Durchläufen wurden nach und nach noch mehr parallele Stationen in das Modell übernommen bis der Durchfluss für eine Laufzeit von ca. 7 Jahren gewährleistet war. Das endgültige Modelldesign enthielt nun folgende parallele Arbeitsstationen:

- Stationen 4, 12, 15,27: drei parallele Arbeitsstationen
- Stationen 11, 19, 21: zwei parallele Arbeitsstationen

Bei Station 11 treffen drei verschiedenen Fertigungslinien wieder zusammen. Dabei können Aufträge von jeder der Linien in jede der beiden Stationen 11 übergeben werden. Bei den beiden aufeinanderfolgenden Stationen 11 und 12 wurden ebenfalls zwei parallele Arbeitsstationen mit drei parallelen Arbeitsstationen gekoppelt. Dabei ist es möglich, von jeder Station 11 den Auftrag in jede Station 12 weiterzusenden.

Nachdem sichergestellt war, dass alle Entitäten korrekt durch das Modell bewegt und die Bearbeitungszeiten korrekt übernommen werden sowie auch der Durchfluss gewährleistet ist wurde das statechart für den Ausfall von Arbeiterentitäten dazugenommen. Dazu wurde monatlich bevor das statechart in den nächsten Zustand wechselte die kumulierten Ausfallszeiten aller Arbeiterarten in das .log- File ausgegeben. Diese Statistik über das ganze Jahr wurde dann mit den tatsächlichen Ausfallszeiten des letzten Geschäftsjahres verglichen. Mittels Korrekturfaktoren für alle Arbeiterarten wurde versucht,

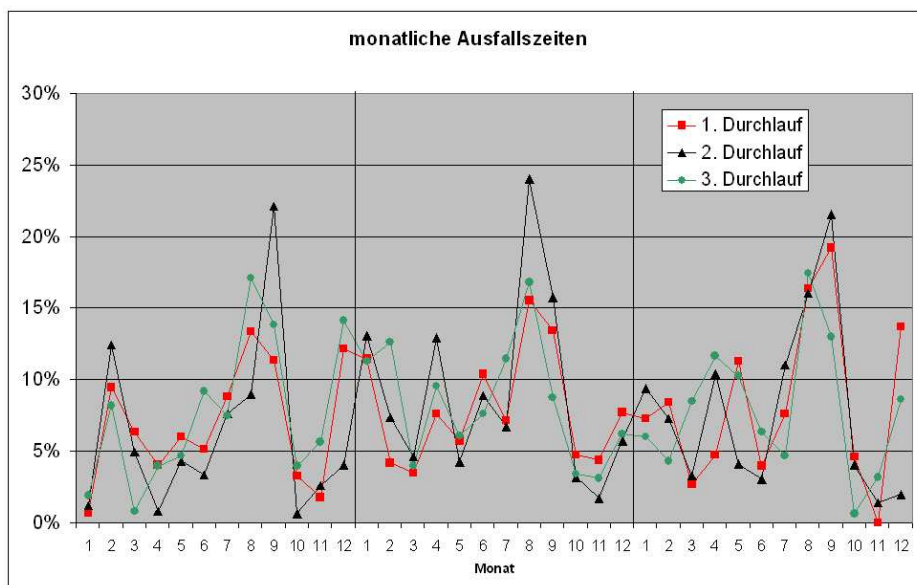


Abbildung 5.13: Anteil der Ausfallszeit an der Gesamtarbeitszeit

sich an die IST- Werte so gut als möglich anzunähern. Die Korrekturfaktoren wurden in mehreren Simulationsläufen empirisch ermittelt.

Abb. 5.13 zeigt den Anteil der Ausfallszeiten für drei Simulationsläufe. Es ist klar erkennbar, dass die Ausfallszeiten den angegebenen Modelldaten trendmässig folgen. Die Simulationsläufe wurden jeweils dreimal mit verschiedenen Einstellungen für die Dauer von drei Jahren durchgeführt um schließlich zur gewählten Konfiguration aus Abb. 5.13 zu gelangen.

Bei diesem nun vollständig lauffähigen Modell wurde nun noch die Verteilung der Bearbeitungszeiten der verschiedenen Fahrzeugvarianten erstellt. Diese Verteilungen zeigten eine Linksschiefe wie in Abb. 5.14 für den Vario-press. Diese Verteilung konnte mit realen Daten verifiziert werden. Auch diese weisen eine leicht linksschiefe Verteilung auf. Dies ergibt sich in der Praxis daraus, dass ein Großteil der Aufträge ähnlich ist und damit auch ähnliche Fertigungszeiten aufweisen. Einige Aufträge weisen jedoch eine Sonderausstattung auf, die entsprechend länger in der Fertigung benötigt. Vereinzelt kommen auch stark angepasste Sonderfahrzeuge vor die sehr lange Bearbeitungszeiten haben.

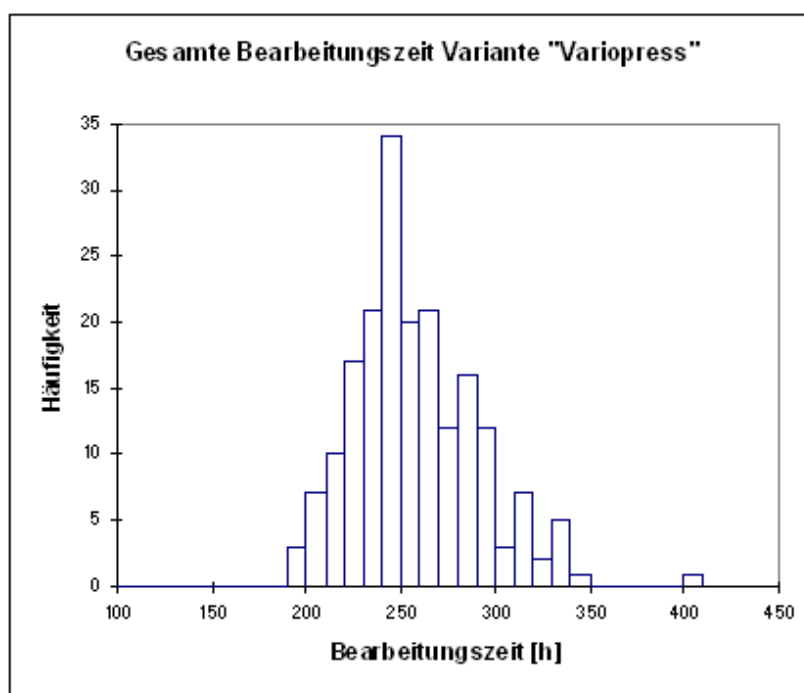


Abbildung 5.14: Verteilung der Bearbeitungszeit beim Typ "Variopress"

Kapitel 6

Untersuchungen

Nachdem das Modell nun lauffähig und validiert ist kann nun mit den Experimenten begonnen werden. Die Simulationsläufe werden dabei solange verändert und angepasst, bis ein praxistaugliches Modell entsteht. Dabei werden die Einstellungen des Modells verändert, ohne jedoch die grundlegende Programmierung oder Funktion zu verändern.

Die Simulationsläufe werden jeweils auf Basis von Zufallsverteilungen durchgeführt. Dabei werden die Ankunftszeiten der Fahrgestelle, Montagezeiten der einzelnen Stationen, Ausfallszeiten usw. nach den jeweiligen Zufallsverteilungen variiert. Jeder Simulationslauf ist damit einmalig und nicht reproduzierbar.

Durch mehrere Simulationsläufe bei sonst gleichen Einstellungen ist es damit möglich, die Praxistauglichkeit der gewählten Konfiguration festzustellen.

AnyLogic bietet dazu die Möglichkeit, verschiedene Experimente ("experiments") anzulegen und zu speichern. Damit kann schnell zwischen verschiedenen Varianten gewechselt werden ohne jedesmal alle Einstellungen neu definieren zu müssen. Neben Simulationsexperimenten können auch Optimierungsexperimente sowie Experimente mit variierenden Parametern durchgeführt werden. Diese bieten sich vor allem für die Analyse und Verbesserung des Modells an.

Der Hauptzweck dieses Modells ist ja wie in Kapitel 1 beschrieben den Produktionsablauf zu optimieren. Besonders hinsichtlich

- Minimierung der gesamten Durchlaufzeit und

- Maximierung der Kapazitätsauslastung

soll eine Verbesserung erzielt werden. Das Modell ist auf Basis eines Jahresplanes aufgebaut, dh. ein Durchlauf des Modells dauert 1 Jahr. Aus Tabelle 2.2 sind auch die tatsächlich gefertigten Stückzahlen des letzten Jahres bekannt und entsprechend den Tabellen 5.1 und 5.2 implementiert. Wird auf Basis dieses Fertigungsplans mit den gegebenen Kapazitäten nun ein Simulationsdurchlauf über ein Jahr durchgeführt der eine höhere Stückzahl bringt so handelt es sich offensichtlich um eine Optimierung hinsichtlich Durchlaufzeit und Kapazitätsauslastung. Dieses Experiment kann mit den Verteilungen aller Fertigungspläne der letzten Jahre oder einer beliebigen anderen Verteilung wiederholt werden.

Da AnyLogic auch die Möglichkeit bietet, Optimierungsexperimente durchzuführen kann in einem weiteren Schritt versucht werden, Durchlaufzeiten und Kapazitätsauslastung noch weiter zu optimieren. Dies sind jedoch gegensätzliche Ziele und bei der Untersuchung wird nur die Verbesserung der Durchlaufzeit behandelt.

6.1 Design der Experimente

Bei jedem Experiment wird eine Vorlaufzeit von 45000 Zeiteinheiten (dh. 45.000 Minuten bzw. 750 Arbeitsstunden) berücksichtigt. Alle bis dahin gesammelten Daten werden für die Auswertung verworfen und nicht berücksichtigt. Diese Vorlaufzeit wird benötigt, um das Modell mit Aufträgen zu füllen und einen Gleichgewichtszustand zu erreichen.

Die weiteren Einstellungen wurden auf folgende Startparameter festgelegt:

- Anzahl Schlosser: 20
- Anzahl Schweißer: 4
- Anzahl Elektriker: 7
- Anzahl Lackierer: 4
- Anzahl Stapler: 2

Für die Experimente werden jeweils zehn Simulationsläufe mit einer Laufzeit von 119.040 Zeiteinheiten (entspricht 248 Arbeitstagen) durchgeführt.

6.2. DURCHFÜHRUNG UND AUSWERTUNG DER EXPERIMENTE 93

	Roto Classic	Roto Styling	Roto ABRA	Vario	Vario SKD
Mittelwert: Anzahl	37,1	20,6	8,3	54,5	63,7
Mittelwert: Bearbeitungszeit	147,7	190,96	210,8	240,48	81,82
Mittelwert: Durchlaufzeit	329,59	410,78	421,66	420,47	243,96
Min.Wert: Anzahl	28	12	6	50	56
Min.Wert: Bearbeitungszeit	113,07	94,54	153,83	199,47	71,67
Min.Wert: Durchlaufzeit	257,7	190,74	303,54	348,56	169,91
Max.Wert: Anzahl	47	26	13	63	80
Max.Wert: Bearbeitungszeit	176,77	332,80	374,36	285,14	110,32
Max.Wert: Durchlaufzeit	392,28	723,06	653,18	506,90	293,93

Tabelle 6.1: Fertigungszeiten Experiment 1: Referenzmodell

6.2 Durchführung und Auswertung der Experimente

Für die Simulations- und Optimierungsexperimente werden folgende Einstellungen untersucht:

- Kompletter Simulationsdurchlauf mit Originalmodell (Referenzmodell)
- Optimierung/ Minimierung der eingesetzten Arbeiter
- Optimierung/ Minimierung der Durchlaufzeit
- Erhöhung der Auftragseingänge

Ausgehend vom Originalmodell wird jeweils eine der obigen Spezifikationen simuliert während die anderen Parameter nicht verändert werden. Jedes Modell kann somit direkt mit dem Originalmodell verglichen werden.

Als Referenz wurde zuerst das Originalmodell simuliert. Mit den angegebenen Startparametern wurden zehn Simulationsläufe durchgeführt. Die Auswertung der Simulationsläufe brachte dabei folgende Ergebnisse:

Insgesamt wurden bei den Simulationsläufen im Durchschnitt 184,2 Aufträge abgeschlossen. Der Minimalwert der abgeschlossenen Aufträge lag bei 170 Stück, der Maximalwert bei 200 Stück. Diese Werte sowie die Werte aus 6.1 können mit den Fertigungszahlen der letzten Geschäftsjahre aus Tab. 2.2 verglichen werden. Im Modell werden bei allen Fahrzeugtypen höhere Stückzahlen erreicht. Auch die Gesamtstückzahl des letzten Geschäftsjahres

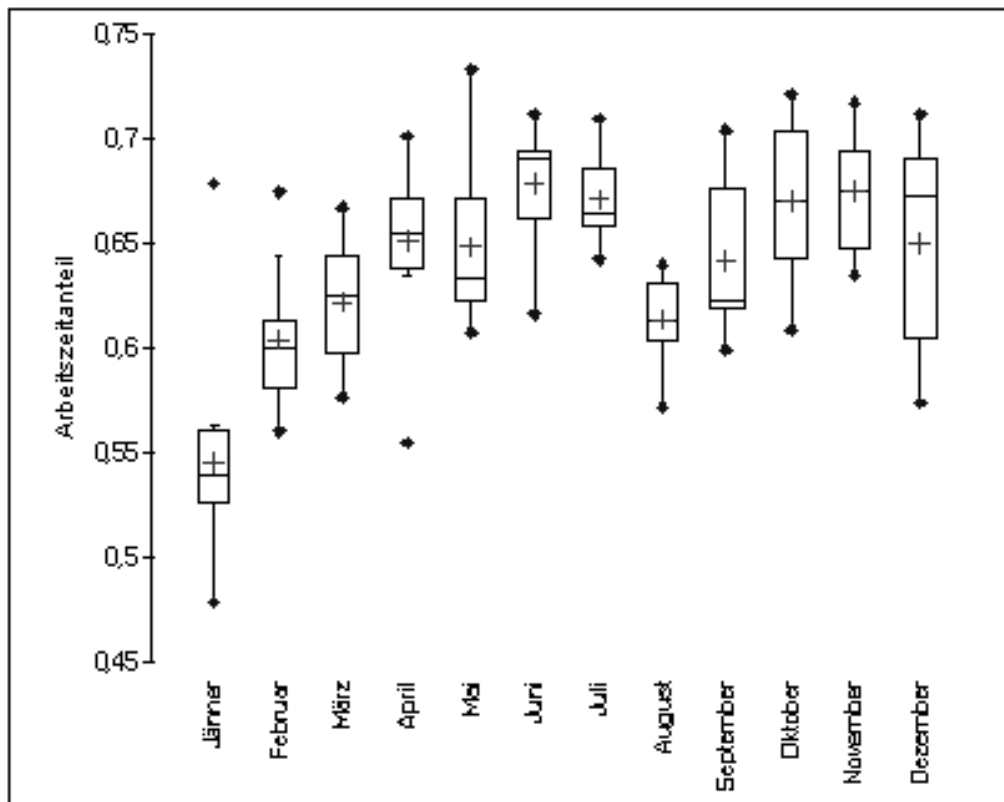


Abbildung 6.1: Experiment 1: Auslastung der Arbeiter

von 125 Stück wird bei weitem übertroffen. Somit ist die flexible Serienfertigung mit Arbeitsstationen bei gleichen Bearbeitungszeiten und gleichem Personalstand dem derzeitigen Fertigungssystem überlegen.

Die Auslastung ist wie die Box-plots in Abb. 6.1 zeigen noch nicht optimal. Im besten Fall liegt die mittlere Auslastung der Arbeiter bei 67% und sinkt auf bis zu 54% ab. Die übrige Zeit ist Wartezeit bzw. unproduktive Zeit oder auch Ausfallszeit wie in Abb. 6.2 ersichtlich. Die mittleren Ausfallszeiten der Simulationsläufe liegen mit Ausnahme der Monate Jänner und Oktober über den Vorgabezeiten oder nur knapp darunter. Die Ausfallszeiten aller drei Simulationsläufe folgen trendmäßig der Vorgabe, ist aber im Monat August stark überhöht. Auf die Produktivität hat das wenig Einfluss da der Ausfall zu Lasten der unproduktiven Zeit geht. Somit besteht hier durchaus noch Optimierungspotential durch Verringerung der Kapazitäten. Das wurde im zweiten Experiment untersucht.

Die Minimierung der Arbeiteranzahl wurde durch Optimierungsexperimente ermittelt. AnyLogic bietet diese die Möglichkeit für diese spezielle

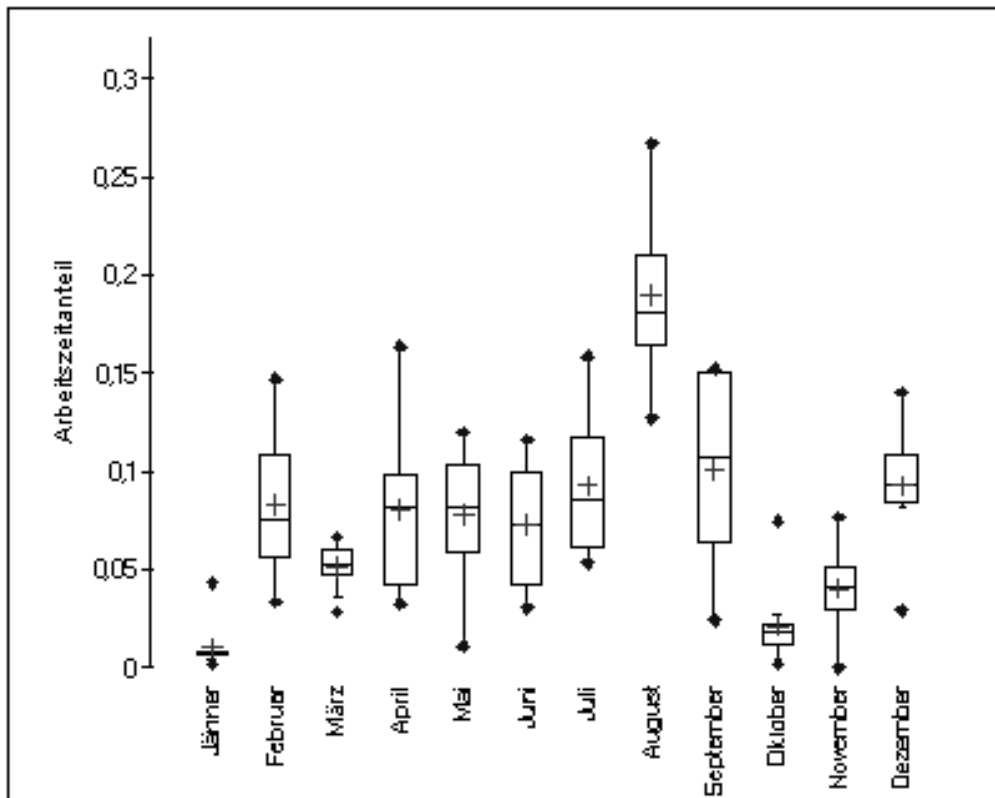


Abbildung 6.2: Experiment 1: Ausfallszeiten der Arbeiter

Form der Simulation. Dabei wird eine Zielfunktion festgelegt die minimiert bzw. maximiert werden soll sowie Parameter die verändert werden können. Weiters wird die Bandbreite, in der die Parameter verändert werden sollen sowie der Startwert der Parameter festgelegt. AnyLogic führt dann eine Reihe von Simulationen mit verschiedenen Einstellungen durch um den Zielfunktionswert zu minimieren. Die Experimente werden dabei so lange wiederholt bis die eingestellte Genauigkeit oder die definierte Anzahl der Durchläufe erreicht ist. Die Auswertung der Experimente kann in AnyLogic selbst graphisch dargestellt oder die Daten nach MS Excel zur weiteren Bearbeitung exportiert werden.

Folgende Arbeiteranzahl wurde dabei als Minimum gefunden, bei dem das Modell noch lauffähig bleibt:

- Anzahl Schlosser: 16
- Anzahl Schweißer: 3
- Anzahl Elektriker: 6
- Anzahl Lackierer: 4
- Anzahl Stapler: 2

Lackierer und Staplerfahrer sind schon am Minimum und können nicht weiter reduziert werden. Die anderen drei Arbeitergruppen könnten reduziert werden. Bei den Schlossern können bis zu vier Arbeiter, bei den Schweißern und Elektrikern jeweils ein Arbeiter gestrichen werden. Im Vergleich zum Referenzmodell sinkt dabei der Output um ca. 4,8% bzw. 9 Fahrzeuge auf 175,3 Fahrzeuge wegen höheren Bearbeitungs- und Durchlaufzeiten. Die Erhöhung der Durchlaufzeit ist dabei bei den Produkten der Rotopress- Baureihe signifikant. Roto Classic (+41,3%) und Roto Styling (+33,6%) stellen einen großen Teil der Produktionsmenge, Verzögerungen fallen damit stärker ins Gewicht (vgl. Tab. 6.1 und Tab. 6.2). Entsprechend geht auch die Produktivität von 188 auf 176 Fahrzeuge bzw. 6,4% zurück.

Wie in Abb. 6.4 ersichtlich bleiben die Ausfallszeiten anteilig in etwa gleich und liegen fast immer über den Vorgabewerten. Der Monat August reisst im Vergleich zu Abb. 6.2 nicht mehr so stark aus. Insgesamt sind in den Sommermonaten Juli bis September sind die Werte stark überhöht und liegen bis zum Vierfachen über den Vorgabewerten. Die Ausfallfaktoren wurden entsprechend minimiert, trotzdem bleiben die Ausfallszeiten zu hoch. Da die Arbeiterzahl schon sehr gering ist entspricht der Ausfall von sehr wenig

6.2. DURCHFÜHRUNG UND AUSWERTUNG DER EXPERIMENTE 97

	Roto Classic	Roto Styling	Roto ABRA	Vario	Vario SKD
Mittelwert: Anzahl	32,9	20	7,6	50,9	63,9
Mittelwert: Bearbeitungszeit	217,17	277,58	312,79	260,02	83,85
Mittelwert: Durchlaufzeit	465,57	548,78	671,11	466,59	252,02
Min.Wert: Anzahl	27	9	2	41	53
Min.Wert: Bearbeitungszeit	196,8	195,42	66,47	214,28	64,66
Min.Wert: Durchlaufzeit	360,66	338,51	382,38	183,5	134,24
Max.Wert: Anzahl	36	25	12	55	79
Max.Wert: Bearbeitungszeit	268,76	560,24	740,27	286,68	106,4
Max.Wert: Durchlaufzeit	581,89	1002,41	1632,19	591,67	393,75

Tabelle 6.2: Fertigungszeiten Experiment 2: Kapazitätsminimierung

Arbeitern schon einem wesentlich höheren als dem zulässigen Wert. Hier wird es schwierig, sich mit der Modellierung der Ausfallzeiten noch an die realen Zahlen anzunähern. Für die Sommermonate kann hier mit einer Deckelung der Arbeiter auf Urlaub noch steuernd eingegriffen werden.

Durch die Kapazitätsminimierung stieg aber auch die Auslastung der Arbeiter wie in Abb. 6.3 ersichtlich. Während im Referenzmodell die durchschnittliche Auslastung bei maximal 67% liegt kann durch die Kapazitätsminimierung die mittlere Auslastung auf bis zu 81% erhöht werden. Die mittlere Auslastung der Urlaubsmonate liegt dabei in Höhe der im Referenzmodell ermittelten maximalen Auslastung von durchschnittlich 67%. In den Urlaubsmonaten ist die Streuung nach unten durch einzelne Ausreisser sehr groß, die Quantile selbst bleiben jedoch weitgehend stabil.

Eine verbesserte Auslastung der Arbeiter geht damit zu Kosten einer erhöhten Durchlaufzeit. Das kann zu Lieferverzögerungen führen was eine entsprechend exakte Planung mit langen Vorlaufzeiten notwendig macht.

Die genau gegenteiligen Einstellungen wurden im dritten Experiment untersucht. Hier wurde versucht, die Durchlaufzeit zu minimieren. Dafür müssen genügend Arbeiter zur Verfügung stehen um die Bearbeitung sofort beginnen zu können. Die Arbeiteranzahl wurde soweit erhöht, dass in jeder Arbeitsstation immer die notwendige Anzahl an Arbeitern verfügbar ist. Das entspricht dem Fall der klassischen Fließproduktion, bei denen jede Arbeitsstation fixe Arbeiterkapazitäten zugeteilt bekommt. Die Einstellungen wurden wie folgt geändert:

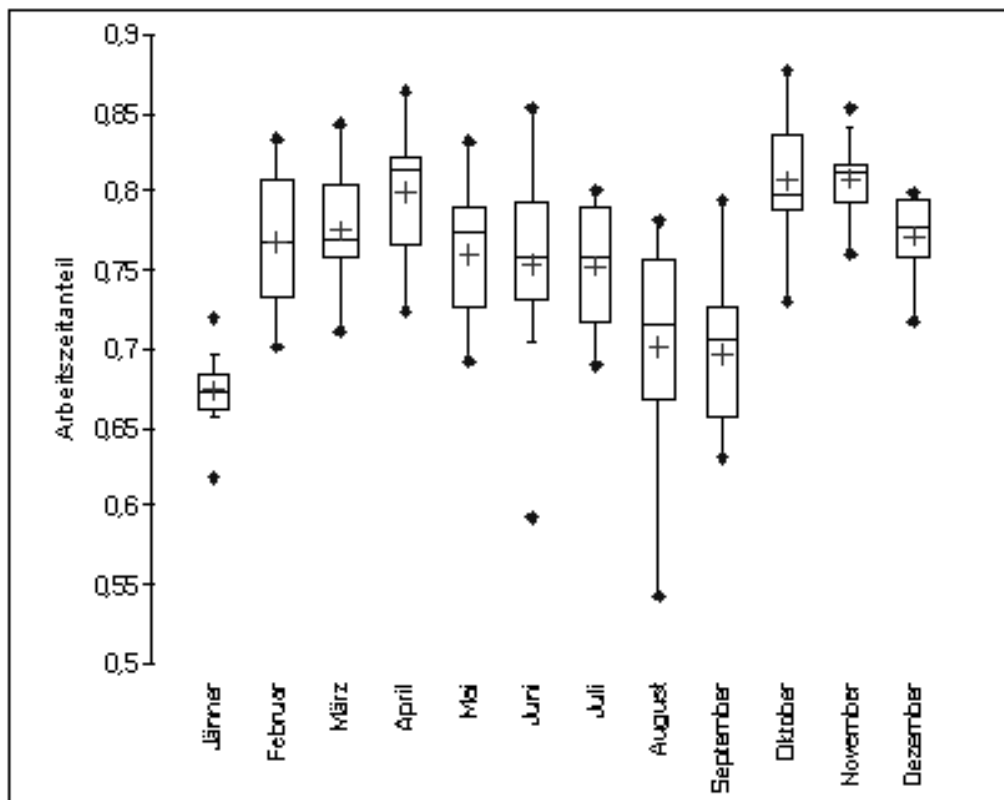


Abbildung 6.3: Experiment 2: Auslastung der Arbeiter

6.2. DURCHFÜHRUNG UND AUSWERTUNG DER EXPERIMENTE 99

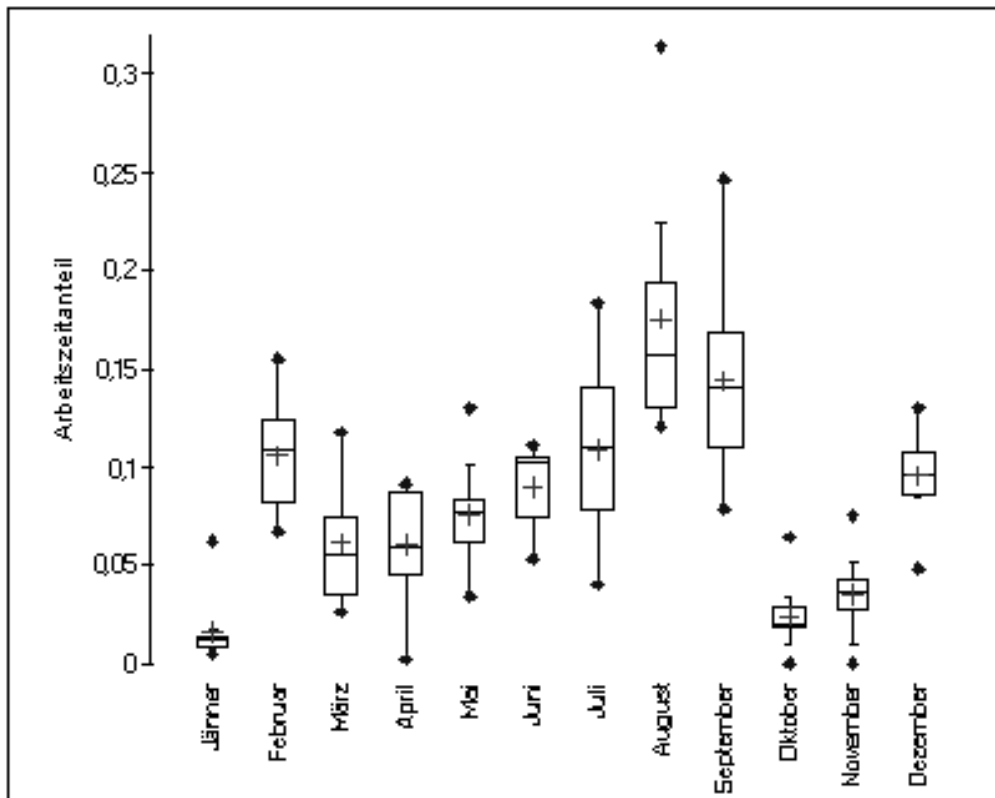


Abbildung 6.4: Experiment 2: Ausfallszeiten der Arbeiter

- Anzahl Schlosser: 26
- Anzahl Schweißer: 12
- Anzahl Elektriker: 15
- Anzahl Lackierer: 4
- Anzahl Stapler: 2

Die Ausfallswahrscheinlichkeiten wurden nicht verändert. Arbeiter können nach wie vor ausfallen. Durch die überhöhte Anzahl von Arbeitern ist die Verfügbarkeit sichergestellt. Durch die Kapazitätserhöhung steigt die Produktivität von durchschnittlich 184 auf 190 Fahrzeuge (+3,5%) an. Dieser Anstieg steht allerdings in keiner Relation zu der zusätzlich eingesetzten Kapazität. Die Arbeiterzahl wurde dafür von 37 auf 59 Arbeiter (+59%) gesteigert. Gleichzeitig ging die mittlere Auslastung der Arbeiter wie in Abb. 6.5 ersichtlich auf 31-40% zurück. Die Bearbeitungszeiten gingen bei allen Fahrzeugtypen leicht zurück (siehe Tab. 6.3). Die Bearbeitungszeit ist in diesem Modell die Verweilzeit in einer Arbeitsstation im Status "Bearbeitung". Wenn verschiedene Arbeiterentitäten in der selben Arbeitsstation einen Auftrag bearbeiten und dabei nicht parallel arbeiten verlängert sich die Verweilzeit. Bei ausreichend Arbeitern und entsprechend paralleler Bearbeitung verkürzt sich die Bearbeitungszeit. Dieser Effekt lässt sich hier beobachten, hat aber nur geringe Auswirkungen. Die Durchlaufzeiten verändern sich ebenfalls nur gering in beiden Richtungen. Auffällig ist hier nur die starke Verringerung der Durchlaufzeit beim Produkt Vario SKD die von 243,96 auf 192,16 Stunden um 21,2% sinkt.

Insgesamt ist keine starke Veränderung der Ergebnisse im Vergleich zum Referenzmodell festzustellen. Eine Steigerung der Arbeiter bewirkt nur eine Verminderung der Arbeitsleistung, hat aber keinen Einfluss auf die Durchlaufzeit. Das Referenzmodell ist also den beiden bisherigen Variationen überlegen und liefert bessere bzw. gleich gute Ergebnisse. Damit scheint sich das Referenzmodell bereits nahe an der optimalen Lösung für die gewählten Einstellungen zu befinden.

Als letzte Variation wird nun angenommen, dass die Anzahl der Auftragseingänge steigt. Dazu wurde die Verteilung der Auftragseingänge verändert, die Anzahl der Arbeiterentitäten bleibt gleich. Die ursprüngliche Verteilung lautete

$$f(\text{Auftrag}) = \max(60, N(462, 180))$$

6.2. DURCHFÜHRUNG UND AUSWERTUNG DER EXPERIMENTE101

	Roto Classic	Roto Styling	Roto ABRA	Vario	Vario SKD
Mittelwert: Anzahl	37,9	21,9	6,9	57,6	66,4
Mittelwert: Bearbeitungszeit	155,92	190,09	218,4	252,45	77,53
Mittelwert: Durchlaufzeit	323,10	377,42	428,04	423,05	192,16
Min.Wert: Anzahl	31	16	4	48	59
Min.Wert: Bearbeitungszeit	127,39	138,15	132,8	203,22	62,07
Min.Wert: Durchlaufzeit	291,03	252,64	255,94	317,74	134,39
Max.Wert: Anzahl	45	29	10	68	78
Max.Wert: Bearbeitungszeit	177,64	292,87	450,41	300,42	99,41
Max.Wert: Durchlaufzeit	359,10	750,65	910,13	528,67	263,07

Tabelle 6.3: Fertigungszeiten Experiment 3: Durchlaufzeitminimierung

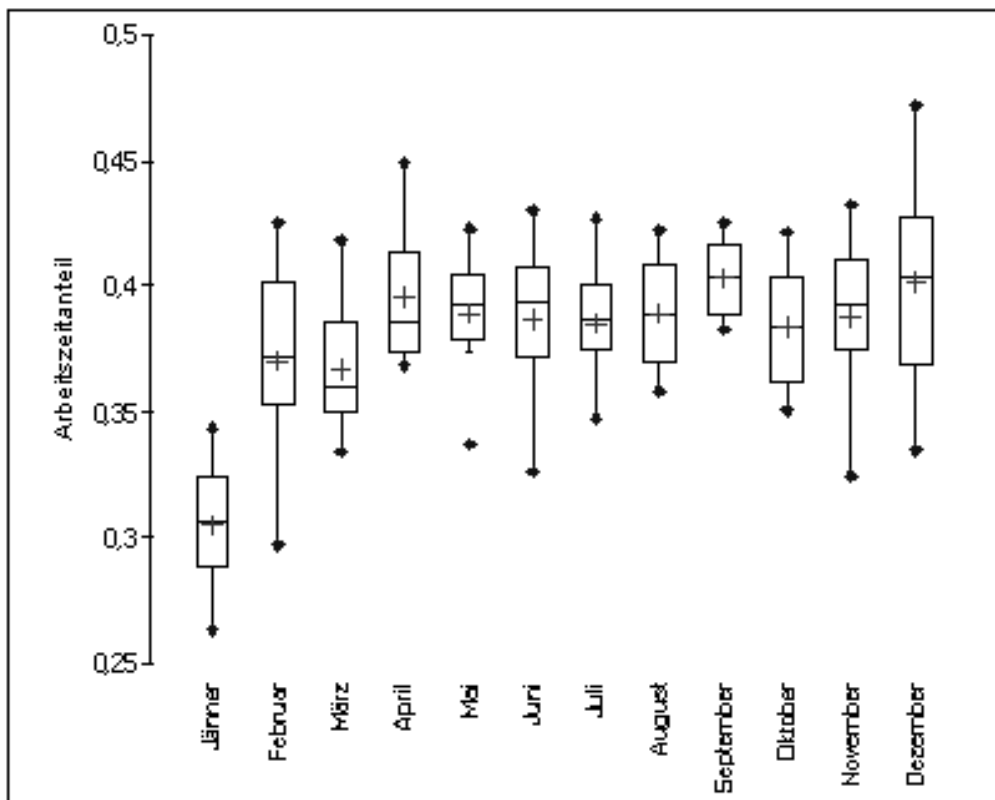


Abbildung 6.5: Experiment 3: Auslastung der Arbeiter

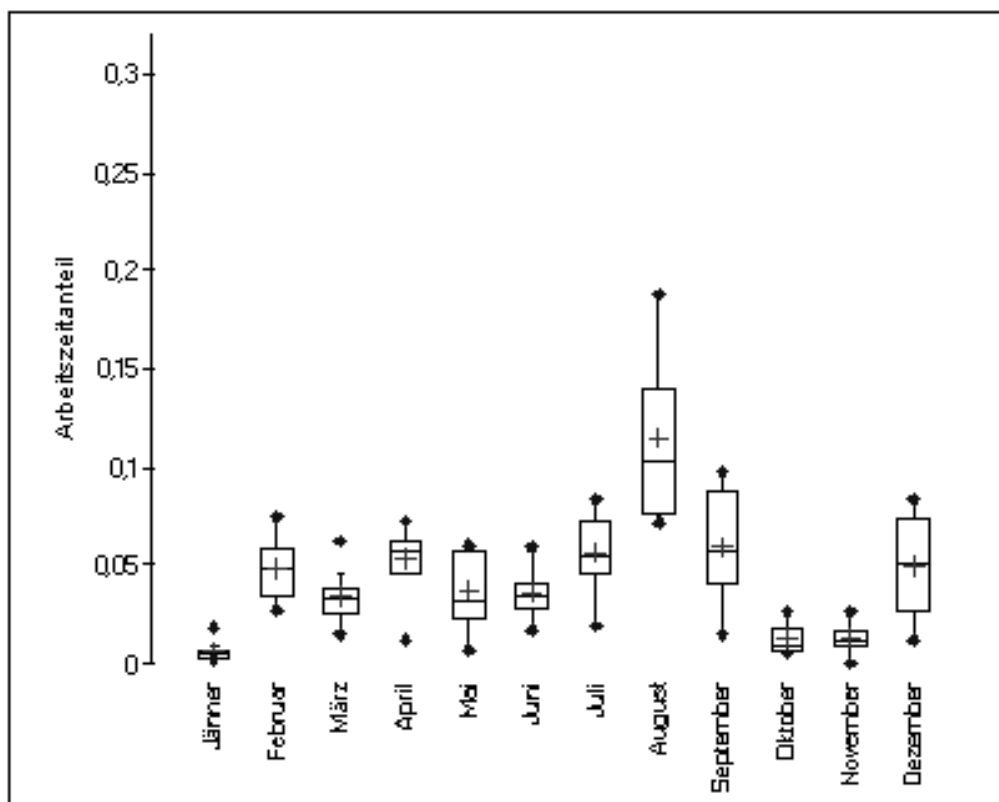


Abbildung 6.6: Experiment 3: Ausfallszeiten der Arbeiter

6.2. DURCHFÜHRUNG UND AUSWERTUNG DER EXPERIMENTE¹⁰³

	Roto Classic	Roto Styling	Roto ABRA	Vario	Vario SKD
Mittelwert: Anzahl	39,9	19,8	7,8	56,9	66,7
Mittelwert: Bearbeitungszeit	168,2	182,19	285,32	255,8	78,8
Mittelwert: Durchlaufzeit	467,91	463,4	710,47	548,12	269,66
Min.Wert: Anzahl	30	16	5	49	54
Min.Wert: Bearbeitungszeit	133,87	101,53	167,74	206,23	60,81
Min.Wert: Durchlaufzeit	339,95	228,49	412,41	424,46	154,29
Max.Wert: Anzahl	49	31	67	82	13
Max.Wert: Bearbeitungszeit	210,36	230,17	492,78	297,5	100,58
Max.Wert: Durchlaufzeit	562,96	605,76	1211,46	677,22	336,81

Tabelle 6.4: Fertigungszeiten Experiment 4: Auftragsmaximierung

Durchschnittlich kommt ein Auftrag je Arbeitstag (7,7 Stunden) in die Fertigung. Der Abstand zwischen zwei Aufträgen beträgt mindestens eine Stunde (60 Minuten). Diese Annahme wurde nun verändert zu

$$f(\text{Auftrag}) = \max(60, N(400, 160))$$

Damit sollte jetzt im Durchschnitt alle 7 Stunden ein Auftrag einlangen und auch die Varianz wurde verringert. Die untere Grenze von einer Stunde zwischen dem Auftragsengang zweier Aufträge wurde beibehalten. Da nun mehr Aufträge in das System kommen steigt wie Tab. 6.4 zeigt der Output auf durchschnittlich 191 Fahrzeuge um 3,7% an. Die Bearbeitungszeiten werden allerdings um 9,5% länger und die Durchlaufzeit steigt überproportional um 32,8% an. Damit wird der Rückstau in den Pufferspeichern ebenfalls länger. Das Modell wird bei längerer Durchlaufzeit instabil, da die Pufferspeicher ausgelastet sind. Längerfristig gesehen kann die höhere Produktivität also mit dem derzeitigen Personalstand und dem derzeitigen Produktionsablauf nicht gehalten werden.

Die Auslastung der Arbeiter steigt im Vergleich zum Referenzmodell wie in Abb. 6.7 ersichtlich ebenfalls leicht an. Auf Grund der erhöhten Durchlaufzeiten zeigt sich allerdings, dass die zur Verfügung stehende Kapazität nicht ausreicht, die in das Modell eingehenden Aufträge abzuarbeiten. Daher wäre eine Kapazitätserweiterung notwendig.

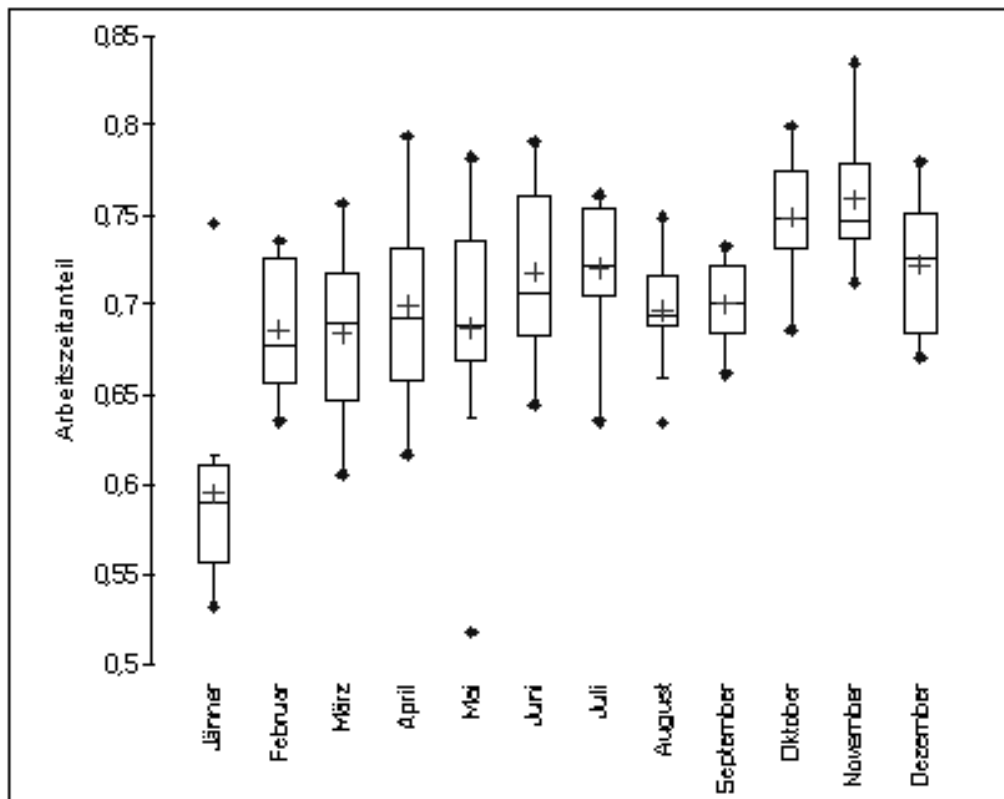


Abbildung 6.7: Experiment 4: Auslastung der Arbeiter

6.2. DURCHFÜHRUNG UND AUSWERTUNG DER EXPERIMENTE105

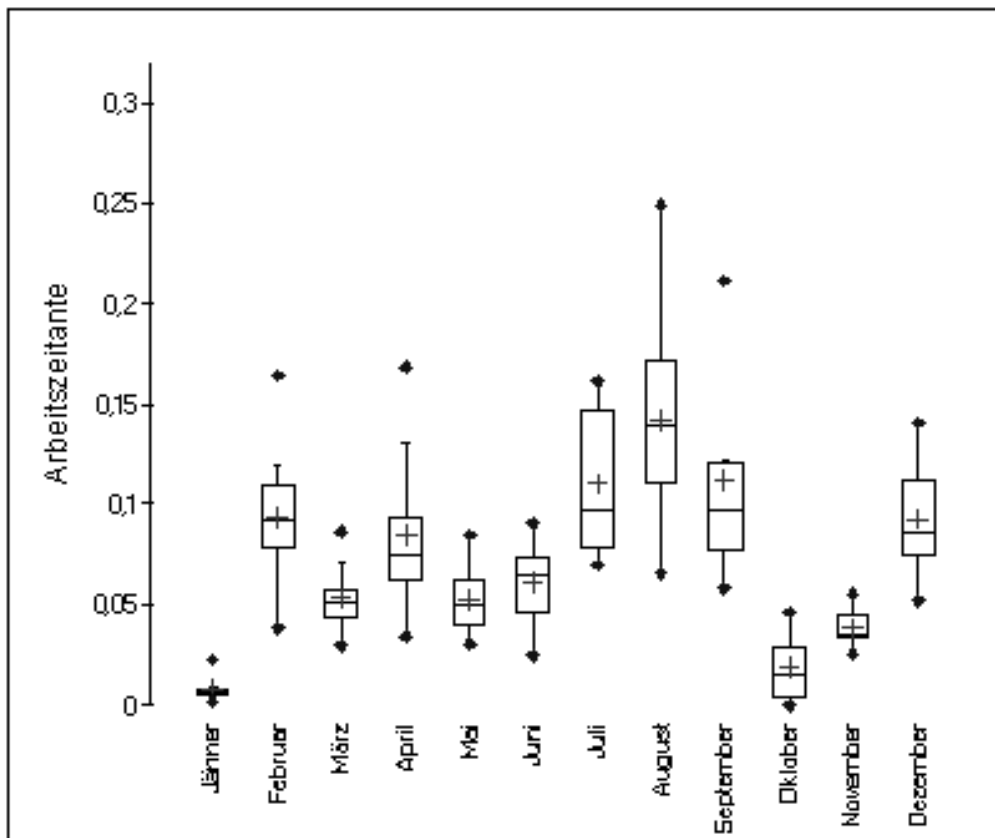


Abbildung 6.8: Experiment 4: Ausfallszeiten der Arbeiter

	Roto Classic	Roto Styling	Roto ABRA	Vario	Vario SKD
Ist-Zeiten GJ 2007/08	491,4	688,72	770	791,32	313,87
Referenzmodell	329,59	410,78	421,66	420,47	243,96
Minimierte Kapazität	465,57	548,78	671,11	466,59	252,02
Minimierte Durchlaufzeit	323,10	377,42	428,04	423,05	192,16
Erhöhter Auftragseingang	467,91	463,40	710,47	548,12	269,66

Tabelle 6.5: Vergleich der mittleren Durchlaufzeiten

6.3 Untersuchungsergebnisse und Diskussion

Wie im vorigen Abschnitt gezeigt ist das Referenzmodell nahe an einer optimalen Lösung. Eine Kapazitätsminimierung ist nicht empfehlenswert da dadurch auch der Output sinkt und Bearbeitungs- und Durchlaufzeiten überproportional steigen. Einziger Vorteil ist eine höhere Auslastung der Arbeiter. Andererseits hat auch eine drastische Kapazitätserhöhung keinen Einfluss auf den Output sowie die Zeiten, senkt aber die Auslastung der Arbeiter wesentlich. Arbeiteranzahl und Outputgrößen stehen also in keiner direkt proportionalen Abhängigkeit.

Tab. 6.5 zeigt nochmals eine Übersicht der realen Durchlaufzeiten zu den in den Simulationsläufen errechneten durchschnittlichen Durchlaufzeiten. Jedes der untersuchten Modelle ist dabei der gegenwärtigen Realsituation mit Einzelplatzfertigung überlegen. Wie in der Tabelle zu sehen ist, ist das Referenzmodell mit den Durchlaufzeiten ähnlich dem Modell mit minimierten Durchlaufzeiten. Die Simulation mit minimierten Kapazitäten bringt ähnliche Ergebnisse wie das Experiment mit erhöhtem Auftragseingang. Obwohl dies die beiden ungünstigsten Modelle darstellen liegen die Ergebnisse immer noch besser als die derzeitige Realsituation.

Wesentlichen Einfluss auf das gesamte Modell haben die Fertigungszeiten der einzelnen Stationen. Auf Grund der stark unterschiedlichen Bearbeitungsdauern kommt es zu Engpässen und Überläufen in den Pufferspeichern. Diese müssen mit parallelen Arbeitsstationen gelöst werden. Dabei müssen bis zu drei Stationen parallel eingerichtet werden um den Durchfluss sicherzustellen. Hier ist eine genauere Planung der Arbeitsstationen wesentlich zielführender als die Anpassung der Arbeiterkapazitäten. Eine andere Zuteilung der Arbeitsschritte auf die Arbeitsstationen könnte dabei die Bearbeitungsdauern glätten und die Produktivität weiter steigern.

Im Modell konnten die Ausfallszeiten nicht genau genug an die realen Vorgaben angepasst werden. Die Ausfallszeiten folgen dem vorgegebenen Trend, liegen aber teilweise weit über den gemessenen Werten. Hier kann kaum mehr eine Anpassung über die Ausfallswahrscheinlichkeit erfolgen da die Korrekturfaktoren schon sehr niedrig sind. Bei sehr wenig Entitäten (beispielsweise nur noch 4 Elektrikern) bewirkt der Ausfall eines Elektrikers einen Rückgang der Produktivität um 25%. Mittels Anpassung der Korrekturfaktoren kann keine Senkung mehr erreicht werden, die nächste Stufe wäre kein Ausfall und damit 0%. Auffällig ist auch, dass die für die Sommermonate Juli und August geplanten höheren Ausfallszeiten sich noch bis in den September hineinziehen obwohl die Ausfallswahrscheinlichkeit schon wieder gesenkt wird. Das kann damit erklärt werden, dass Ende August noch vergleichsweise viele Arbeiter einen Urlaub beginnen, der Ausfall selbst allerdings erst im September wirksam wird. Bei allen Arbeitern wurde weiters eine Deckelung der maximalen Ausfälle im Modell vorgesehen. Diese Deckelung lag je Arbeiterart bei maximal 33% der verfügbaren Arbeiter die gleichzeitig ausfallen durften. Diese Grenze kann noch weiter gesenkt werden, führt aber zu ähnlichen Problemen wie bei der Steuerung über die Aufallsfaktoren.

Insgesamt hätte die reale Umsetzung noch einige Reserven übrig. Diese sind für die operative Planung sowie Feinabstimmung auch notwendig. Das Modell ist mit der derzeitigen Auftragsrate (im Schnitt 1 Auftrag/ 462 Minuten) auch gut ausgelastet. Da der derzeitige Output leicht unter dem Input liegt sind alle durchgeführten Simulationen langfristig instabil. Eine drastische Erhöhung des Auftragseinganges ist mit dem derzeitigen Modell nicht zu bewältigen.

Im Vergleich zu den zur Verfügung stehenden Daten kann das Modell die derzeitigen Auftragseingänge mit einer Rate die unter den Auftragseingängen im Modell liegen jedoch bewältigen. Dabei wird auch weniger Kapazität gebunden als beim derzeitigen Montageablauf in der M-U-T bei gleichzeitig kürzerer Durchlaufzeit. Die Bearbeitungszeiten bewegen sich auf dem selben Niveau.

Kapitel 7

Zusammenfassung und Ausblick

Auf Grund der gegebenen Problemstellung war die Umsetzung als Simulationsmodell der richtige Ansatz. Die Komplexität der Variantenfertigung sowie die speziellen Gegebenheiten im gewählten Betrieb hätten eine Umsetzung mit anderen mathematischen Modellen wesentlich erschwert bzw. gar nicht möglich gemacht.

Simulationsmodelle bieten eine sehr gute Unterstützung bei der Engpaßfindung bei unterschiedlichen Bedingungen (aktuelle Belastung der Produktion, aktuelle Aufträge), ihre Anwendung ist aber auch sehr sinnvoll bei der Überprüfung des Nutzens von verschiedenen Verbesserungsmaßnahmen. [KG95], S.93

Dieses Modell bietet noch keine konkrete Lösung für die Layoutplanung des Endmontagebereiches. Hier sind andere räumliche Faktoren zu ermitteln und in eine Layoutplanung einzubringen. Einen Ansatz zur Layoutplanung bei stark montagelastigen, kundenauftragsorientierten Unternehmen bietet Petersen. Dieser bietet einen heuristischen Algorithmus zur Montageflächenbelegung und -planung (vgl. [Pet92]).

Produktionskosten werden in dem vorliegenden Modell ebenfalls nicht berücksichtigt. Dies wäre mit AnyLogic zwar prinzipiell möglich, würde die tatsächlichen Kosten jedoch nur unzureichend erfassen können. In AnyLogic kann man bei jeder Arbeitsstation, jeder Arbeiterentität sowie jeder Auftragsentität Kosten hinterlegen und diese auch berechnen. Hinsichtlich der planungsrelevanten Kosten wäre diese Kalkulation allerdings unvollständig. Darauf weist auch schon Schneeweiß hin und führt dabei folgende Gründe an (vgl. [Sch02], S94ff):

- Die Dynamik sowie die im Produktionsgeschehen vorhandene Unsicherheit werden nicht berücksichtigt.

- Die ungenügende Einbeziehung der betrieblichen Organisationsstruktur.
- Die nicht ausreichende Berücksichtigung von Führungstätigkeiten.
- Die Beschränkung auf quantitative Größen.
- Dienstleistungen werden nur sehr ungenügend erfasst.
- Produktionskoeffizienten, welche nur über Aggregationsverfahren ermittelt werden können.

Aus all jenen Gründen wäre eine Kalkulation der Kosten nur unzureichend gewesen. Schneeweiß spricht dabei zwar hauptsächlich von Fertigungsgemeinkosten, tatsächlich würden auch Fertigungseinzelkosten nicht berücksichtigt.

Strategische Investitionsplanung und Wirtschaftlichkeitsrechnung von FFS-Systemen werden von Wildemann/ Bühner ausführlich behandelt. [HW87] Diese fassen den FFS- Begriff jedoch etwas weiter sodass sich diese Berechnungen auch auf Produktionsinseln sowie Variantenfließfertigung anwenden lassen.

Literaturverzeichnis

- [Asp00] ASPRION, B.: *Produktionsplanung bei kundenauftragsorientierter Kleinserienproduktion*. Service Fachverlag, Wien, 2000
- [CC08] CHEN, J. ; CHEN, F.F.: Adaptive scheduling and tool flow control in flexible job shops. In: *International Journal of Production Research* 46 (2008), Nr. 15. – Online unter <http://www.informaworld.com/10.1080/00207540701197002>, abgerufen am 17.07.2008
- [CDS70] CAMPBELL, H. G. ; DUDEK, R. A. ; SMITH, M. L.: A Heuristic Algorithm for the n Job, m Machine Sequencing Problem. In: *Management Science* 16 (1970), Nr. 10, S. B630–B637. – Online unter <http://www.jstor.org/pss/2628231>, abgerufen am 07.08.2008
- [CMT06] COLLEDANI, M. ; MATTA, A. ; TOLIO, T.: Performance evaluation of production lines with finite buffer capacity producing two different products. In: LIBEROPOULOS, G. (Hrsg.): *Stochastic Modeling of Manufacturing Systems*. Springer, Berlin Heidelberg, 2006
- [DMA08] DROELT, J. ; MARCOUX, Y. ; ABDULNOUR, G.: Simulation-based performance comparison between dynamic cells, classical cells and job shops: a case study. In: *International Journal of Production Research* 46 (2008), Nr. 2. – Online unter <http://www.informaworld.com/10.1080/00207540601138312>, abgerufen am 17.07.2008
- [Dom07] DOMSCHKE, W.: *Einführung in Operations Research*. 7. überarb. Auflage. Springer, Berlin, 2007
- [EI93a] E.DICHTL ; ISSING, O.: *Vahlen's Großes Wirtschaftslexikon*. Bd. 2. 2. überarb. und erw. Aufl. Verlag Franz Vahlen, München, 1993

- [EI93b] E.DICHTL ; ISSING, O.: *Vahlens Großes Wirtschaftslexikon*. Bd. 1. 2. überarb. und erw. Aufl. Verlag Franz Vahlen, München, 1993
- [GT95] GÜNTHER, H.O. ; TEMPELMEIER, H.: *Produktionsmanagement*. 2. vollst. überarb. und erw. Auflage. Springer Verlag, Berlin Heidelberg New York, 1995
- [GT05] GÜNTHER, H.O. ; TEMPELMEIER, H.: *Produktion und Logistik*. 6. Auflage. Springer Verlag, Berlin Heidelberg New York, 2005
- [HL04] HILLIER, F.S. ; LIEBERMAN, G.J.: *Introduction to operations research*. 8. McGraw- Hill, New York, 2004
- [HW87] H. WILDEMANN, R.Bühner: *Investitionsplanung und Wirtschaftlichkeitsrechnung für FFS*. Fachverlag für Wirtschaft und Steuern, Schaeffer GmbH & Co, Stuttgart, 1987
- [Kel02] KELTON, W.D.: *Simulation with ARENA*. 2. internat. McGraw-Hill, Boston, 2002
- [KG95] KOŠTURIK, J. ; GREGOR, M.: *Simulation von Produktionssystemen*. Springer Verlag, Wien New York, 1995
- [LG91] LOCKYER, K. ; GORDON, J.: *Critical Path Analysis and other Project Network Techniques*. 5. Pitman Publishing, London, 1991
- [Nah05] NAHMAS, St.: *Production and Operations Analysis*. International Edition, 5. Auflage. McGraw-Hill, New York, 2005
- [Neu75] NEUMANN, K.: *Operations Research Verfahren, Band III*. Carl Hanser Verlag, München Wien, 1975
- [Neu02] NEUMANN, K.: *Operations Research*. Hanser, München Wien, 2002
- [NJON06] NYEN, P.v. ; J.BERTRAND ; OOIJEN, H.P.G.v. ; N.J.VANDAELE: A heuristic to control integrated multi-product multi-machine production-inventory systems with job shop routings and stochastic arrival, set-up and processing times. In: LIBEROPOULOS, G. (Hrsg.): *Stochastic Modeling of Manufacturing Systems*. Springer, Berlin Heidelberg, 2006
- [Ock85] OCKER, H.: *Ein Beitrag zur Produktivitäts- und Flexibilitätanalyse in der Einzel- und Kleinserienfertigung*, Universität Köln, Diss., 1985

- [Pet92] PETERSEN, U.: *Produktionsplanung und Belegung von Montageflächen*. Gabler, 1992
- [Praa] PRAEHOFER, H.: *Sprachen und Modellierungssysteme der diskrete[!] Simulation*. – Vorlesungsskriptum
- [Prab] PRAEHOFER, H.: *Stochastische Simulation*. – Vorlesungsskriptum
- [Sch95] SCHOLL, Armin: *Balancing and Sequencing of Assembly Lines*. Physica Verlag Heidelberg, 1995
- [Sch02] SCHNEEWEISS, Ch.: *Einführung in die Produktionswirtschaft*. 8. Auflage. Springer Verlag, Berlin Heidelberg New York, 2002
- [Swa05] SWAIN, J.J.: Gaming Reality: Biennial survey of discrete-event simulation software tools. In: *OR/MS today* (2005)
- [Swa07] SWAIN, J.J.: New Frontiers in Simulation: Biennial survey of discrete-event simulation software tools. In: *OR/MS today* (2007)
- [Tem91] TEMPELMEIER, H.: *Simulation mit SIMAN*. Physica-Verlag, Heidelberg, 1991
- [Tie92] TIEDT, U.: *Effektivitätssteigerung von FFS durch Simulation und Optimierung der Einlastungsreihenfolge*, Technische Universität Dresden, Diss., 1992
- [TK93] TEMPELMEIER, H. ; KUHN, H.: *Flexible Fertigungssysteme*. Springer Verlag, Berlin Heidelberg New York, 1993
- [VAR06] VUUREN, M.v. ; ADAN, I. ; RESING-SASSEN, S.A.E.: Performance analysis of multi-server tandem queues with finite buffers and blocking. In: LIBEROPOULOS, G. (Hrsg.): *Stochastic Modeling of Manufacturing Systems*. Springer, Berlin Heidelberg, 2006
- [VDI00] VDI: *VDI 3633-Blatt 1: Simulation von Logistik-, Materialfluß- und Produktionssystemen- Grundlagen*. VDI, Düsseldorf, 2000
- [War96] WARNECKE, H.-J.: *Die Montage im flexiblen Produktionsbetrieb*. Springer Verlag, Berlin Heidelberg New York, 1996
- [XJ] XJ TECHNOLOGIES COMPANY LTD. (Hrsg.): *ANYLOGIC User's manual*. XJ Technologies Company Ltd.. – www.xjtek.com/products/anylogic, ©1992-2005

Anhang A

Objekte des AnyLogic- Modells

Anhang A zeigt die Strukturdiagramme sowie Parametereinstellung der selbst erstellten Objektklassen. Jedes aktive Objekt besitzt ein Strukturdiagramm. Dieses erfüllt die Aufgaben (vgl. [XJ], S.20):

- Definition der Schnittstellen
- Definition der Objekte in der neuen Objektklasse sowie deren Verbindungen
- Definition von Zustandselementen wie Timern, Statecharts etc.

Auf Grund des hierarchischen Aufbaus von AnyLogic- Projekten können in jeder Objektklasse auch weitere Objekte verwendet werden. Die Verschachtelung kann beliebig tief erfolgen. Jede Klasse benötigt Ein- und Ausgänge welche im Strukturdiagramm definiert werden. Die Ports können mit den Objekten mittels Konnektoren fix oder mittels Programmcode auch variabel verbunden werden. In diesem Modell sind alle Ports über Konnektoren fix mit den internen Objekten verbunden.

Die Parameter in den Objektklassen dienen dazu, die Einstellungen und Parameter für das Objekt zu setzen. Damit können für verschiedene Instanzen eines Objektes verschiedene Parameterwerte gesetzt werden. Alle Parameterwerte können während der Modellausführung verändert werden. Parameter von aktiven Objekten können auch dazu verwendet werden, eingebaute Objekte zu parametrisieren.

Neben den Parametern in aktiven Objekten und Entitäten gibt es auch global definierte Parameter die für das gesamte Modell gelten (siehe A.1). Globale Parameter werden direkt im Main- Objekt des Modells definiert.

Das hat den Vorteil, dass von allen Objekten innerhalb der Main auf diese Parameter referenziert werden kann. Bei einer Änderung der globalen Parameter werden auch die lokal referenzierten Parameter geändert. Global ist hierbei nur die Anzahl der jeweiligen Arbeiterentitäten relevant. Diese globale Definition hat auch den Vorteil, dass Experimente und Optimierungsläufe in AnyLogic einfach durchgeführt und gespeichert werden können. Bei jedem einzelnen Experiment/ Simulation können die globalen Parameter mit anderen Werten durchgeführt und auch gespeichert werden. Damit sind die genauen Simulationseinstellungen auch später noch bekannt.

Name	Type	Standardwert	Beschreibung
anzahlLackierer	integer	3	Anzahl der Lackierer im Modell
anzahlStapler	integer	2	Anzahl Stapler im Modell
anzahlElektriker	integer	4	Anzahl Elektriker im Modell
anzahlSchlosser	integer	13	Anzahl Schlosser im Modell
anzahlSchweisser	integer	3	Anzahl Schweißer im Modell
vorlaufzeit	integer	45000	Zeit bevor Datenaufzeichnung beginnt

Tabelle A.1: Parameter des Modells

A.1 Aktive Objekte

A.1.1 Objekt "Arbeitersource"

Die Arbeitersource aus Abb. A.1 hat zwei Eingänge. Im Main- Objekt werden an die Eingänge die Eingangsports für Arbeitsaufträge und Arbeiterentitäten angeschlossen (siehe Abb. A.2). Diese externe Anbindung ist notwendig damit die Arbeitersource für alle Arbeiterarten verwendet werden kann. Ansonsten wäre für jede Arbeiterart ein eigenes Source- Objekt notwendig gewesen.

Die Parameter dieses Objekts sind in Tab. A.2 angeführt. Der parameter *anzahl* referenziert auf die entsprechenden globalen Parameter aus Tab. A.1. Arbeiterart und Farbe sind ebenso wie der ausfallfaktor fix eingestellt. Der ausfallfaktor für jede Arbeiterart wurde in der Validierungsphase empirisch ermittelt. Die Vorlaufzeit wird vom Parameter *vorlaufzeit* der Main übernommen. Damit wird die Vorlaufzeit im Statechart gesteuert.

Statecharts werden verwendet ereignis- oder zeitgesteuerte Statusänderung von Objekten zu modellieren und auch zu visualisieren. Ein *statechart* zeigt den Status eines Objektes sowie die Aktionen oder Zeitpunkte zu denen eine

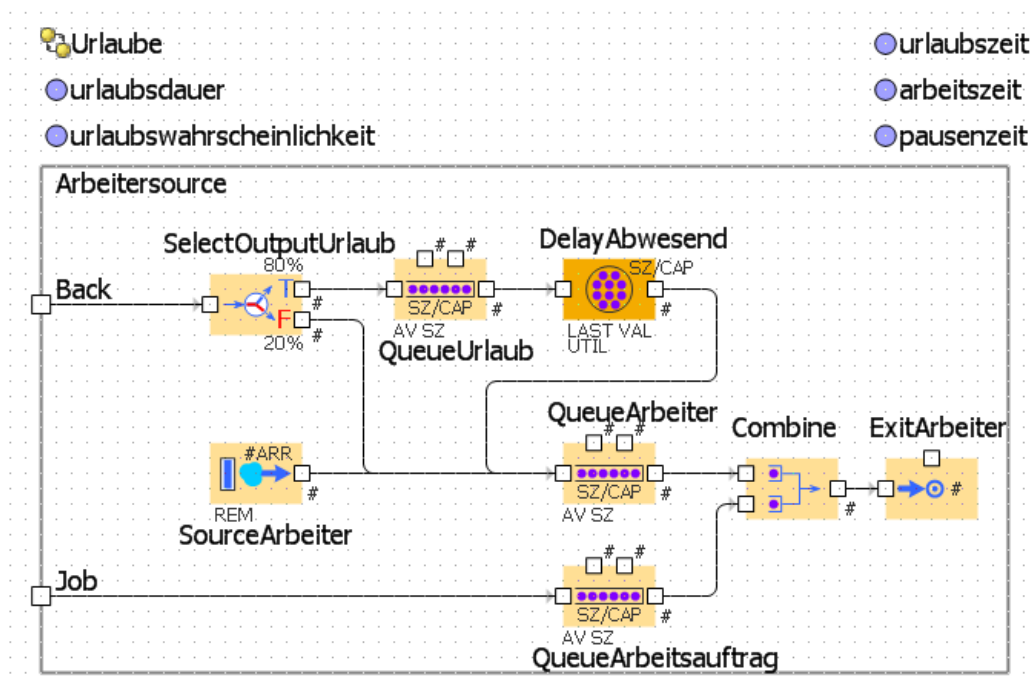


Abbildung A.1: Objekt Arbeitersource

Änderung, dh. der Übergang, zwischen zwei verschiedenen Status erfolgt. Im vorliegenden Modell wird ein statechart dazu verwendet die verschiedenen Urlaubswahrscheinlichkeiten im Jahresverlauf darzustellen. Dabei wird jedes Monat die Urlaubswahrscheinlichkeit neu festgesetzt (vgl. A.3).

A.1.2 Objekt "Bearbeitung"

Dieses Objekt wird für die Darstellung der Arbeitsstationen im Modell verwendet. Wie in Tab. A.3 ersichtlich legen die Parameter *anzSchweisser*, *anzSchlosser* und *anzElektriker* die maximal benötigte Anzahl von Arbeitern fest die in dieser Arbeitsstation notwendig sind. Auf Grund dieser maximalen Arbeiteranzahl wird die entsprechende Anzahl von Arbeitsauftrags- Entitäten generiert. Damit wird sichergestellt, dass immer nur die maximale Anzahl von Arbeitsauftrags- Entitäten für den nächsten Arbeitsauftrag bereitsteht. Das reduziert die summierte Anzahl der Arbeitsauftrags- Entitäten für das gesamte Modell und erhöht damit die Rechengeschwindigkeit. Die *delay*- Parameter werden direkt vor Beginn des entsprechenden Auftrages aus der Datenbank mittels Java- Methode befüllt.

Stationsnr und *stationsaufruf* sind Steuerparameter für das Modell um die

Name	Type	Standardwert	Beschreibung
anzahl	integer		Anzahl der Arbeiterentitäten die in diesem Objekt sind
arbeiterart	String	"Schlosser"	Art der Arbeiterentitäten die mit diesem Objekt verwaltet werden
farbe	String	"blau"	Definiert die Farbe für die Darstellung in der Animation
ausfallfaktor	real	1	Gewichtungsfaktor für die Wahrscheinlichkeit eines Ausfalls der Arbeiterobjekte
maxUrlaube	integer	0	Anzahl Arbeiter die gleichzeitig abwesend sein können

Tabelle A.2: Parameter des Objekts Arbeitersource

Arbeitsauftrags- und Arbeiterentitäten richtig zuordnen zu können.

Im Objekt selbst gibt es nach allen Eingängen Warteschlangen um die Entitäten aufzunehmen (siehe Abb. A.4). In der QueueAuftrag warten Kundenaufträge auf die Bearbeitung. Der Kundenauftrag in der QueueArbeit wird jeweils aktuell bearbeitet und verbleibt solange in der Queue, bis alle Arbeiterentitäten ihre Tätigkeiten abgeschlossen haben. Die hold- Objekte dienen der Steuerung der Entitäten und werden von diesen mittels Programmcode gesperrt oder geöffnet. Das Objekt HoldAuftrag wird entsperrt wenn alle Arbeiterentitäten die entsprechenden Delay- Objekte verlassen haben. HoldFertigungsauftrag wird freigegeben wenn der Kundenauftrag das Objekt QueueArbeit verlässt und HoldArbeitsauftrag wird geöffnet wenn alle Arbeitsauftragsentitäten generiert und parametrisiert sind.

A.1.3 Objekt "Lackierarbeiten"

Dieses Objekt stellt die Lackiererei des Unternehmens mit ihren drei Lackierboxen dar. Die Entitäten sind fix zugeordnet und verlassen dieses Objekt nicht. Anfahrts- und Abholzeit sind ebenfalls fixiert, anzLackierer ist mit dem zugehörigen globalen Parameter verknüpft. Die Parameter dieses Objekts sind in Tab. A.4 angeführt.

Der Bearbeitungsfluss durch dieses Objekt ist linear. Wie in Abb. A.5 ersichtlich besteht es im wesentlichen aus drei hintereinander angeordneten Fertigungsstationen mit jeweils eigenem Pufferspeicher. Die erste und letzte Station wird durch die gemeinsame Resource "Stapler" bedient und stellt den Transport der Teile dar. Die Station "Lackiervorgang" modelliert den

Name	Type	Standardwert	Beschreibung
stationsnr	integer		Fortlaufende Nummerierung für das Objekt im Modell
stationsaufruf	String		Stationsname für das Objekt im Modell
interarrival	integer	100000	Ankunftszeit für die Entitäten "Arbeitsauftrag"
anzSchlosser	integer	0	Anzahl der Schlosser, welcher in diesem Objekt für alle Varianten maximal benötigt wird
anzSchweisser	integer	0	Anzahl der Schweißer, welcher in diesem Objekt für alle Varianten maximal benötigt wird
anzElektriker	integer	0	Anzahl der Elektriker, welcher in diesem Objekt für alle Varianten maximal benötigt wird
delaySchlosser	real	triangular (0.5,1,1.5)	Dreiecksverteilte Arbeitszeit des Schlossers
delaySchweisser	real	triangular (0.5,1,1.5)	Dreiecksverteilte Arbeitszeit des Schweißers
delayElektriker	real	triangular (0.5,1,1.5)	Dreiecksverteilte Arbeitszeit des Elektrikers
queueLength	integer	100	Größe der Warteschlange/Pufferspeicher für die Arbeitsstation

Tabelle A.3: Parameter des Objekts Bearbeitung

eigentlichen Lackierschritt. Hier können bis zu drei Teile gleichzeitig lackiert werden.

A.1.4 Objekt "Gemeinsame Weitergabe"

Dieses Objekt in Abb. A.6 dient nur der Zusammenführung von Kundenauftragsteilen die auf parallelen Arbeitsstationen bearbeitet wurden. Die Entitäten kommen dabei durch die beiden Eingänge in das Objekt und warten in den parallelen Warteschlangen. Wenn in beiden Schlangen Aufträge mit der selben Auftragsnummer warten werden diese im Assembly- Objekt kombiniert und durch den Ausgang weitergesendet.

Name	Type	Standardwert	Beschreibung
anfahrtszeit	integer	25	Zeit, um die Lackierteile von der Montagehalle zur Lackiererei zu befördern
abholzeit	integer	20	Zeit, um die lackierten Teile in die Montagehalle zu befördern
anzLackierer	integer	3	Anzahl der Lackierer
anzStapler	integer	1	Anzahl der Stapler
delayLackierer	real	triangular (0.5,1,1.5)	Dreiecksverteilte Arbeitszeit des Lackierers

Tabelle A.4: Parameter des Objekts Lackierarbeiten

A.2 Entitäten

A.2.1 Kundenauftrags- Entitäten

Die Entität 'Kundenauftrag' stellt die Kundenaufträge dar die im System eintreffen und bearbeitet werden. Die Entitäten werden stochastisch in der Quelle erstellt. Die Parameter *fahrzeugtyp*, *verkleidung* und *lackierschritt* dienen für die Steuerung der Entität. Der Parameter *auftnr* zählt eine fortlaufende Nummer mit die als Auftragsnummer verwendet wird. Alle anderen Parameter dienen der Aufzeichnung von Daten während die Entität durch das Modell läuft. Dies sind hauptsächlich Zeitpunkte und Zeiten sowie der Parameter *nacharbeit*. Dieser protokolliert mit, ob der zusätzliche Arbeitsschritt 'Fehler nacharbeiten' notwendig ist.

A.2.2 Arbeitsauftrags- Entitäten

Die einzige Funktion der Arbeitsauftragsentitäten ist es, Arbeiter für die notwendigen Tätigkeiten in den Arbeitsstationen zu holen. Je nach Wert des Parameters "arbeiterart" wird die Entität an die entsprechende Arbeitersource übergeben. Bei der Zuteilung zu einem Arbeiter wird die Arbeitsauftragsentität vernichtet. Zuvor wird noch der Wert im Parameter "arbeitsstation" an die Entität "Arbeiter" übergeben.

A.2.3 Arbeiter- Entitäten

Diese stellen die verfügbaren Arbeiter dar. Die *arbeiterart* legt fest, um welchen der drei möglichen Arbeitertypen (Schlosser, Schweißer, Elektriker) es

Name	Type	Standardwert	Beschreibung
fahrzeugtyp	integer	1	Zur Auswahl des Fahrzeugtyps für den Auftrag
wartezeit	real	0	Mitprotokollieren der Stehzeiten für den Auftrag
bearbeitungszeit	real	0	Mitprotokollieren der Arbeitszeit für den Auftrag
auftrnr	integer		Die laufende Auftragsnummer für die einzelnen Aufträge
nacharbeit	boolean	false	Um zu markieren ob der Auftrag nachbearbeitet wurde
beginnzeit	integer		Zeitpunkt, zu dem der Auftrag gestartet wird
endzeit	real		Zeitpunkt, zu dem der Auftrag beendet wird
lackierschritt	String	"EnterLackier schritt1"	Zähler für Lackierschritt
verkleidung	integer	1	Codierung für Verkleidungstyp des Fahrzeuges
arbeitszeit	real	0	Zeit, an der effektiv am Fahrzeug gearbeitet wurde

Tabelle A.5: Parameter der Entität Kundenauftrag

sich handelt. Als Steuerparameter wird der Parameter *arbeitsstation* verwendet. Dieser gibt an, wohin sich die Entität zur Ausführung des nächsten Arbeitsauftrages begeben soll. Die Zeit- Parameter speichern die jeweiligen Verwendungszeiten für jede einzelne Entität.

Name	Type	Beschreibung
arbeitsstation	String	Zu welcher Arbeitsstation gehört der Arbeitsauftrag
arbeiterart	String	Welcher Arbeiter benötigt wird

Tabelle A.6: Parameter der Entität Arbeitsauftrag

Name	Type	Beschreibung
arbeitsstation	String	Zu welcher Arbeitsstation gehört der Arbeitsauftrag
arbeiterart	String	Welcher Arbeiter benötigt wird
arbeitszeit	real	Zähler für die tatsächliche Arbeitszeit am Auftrag
wartezeit	real	Zähler für die Zeit, die in Warteschlangen verbracht wurde
urlaubszeit	real	Zähler für die Zeit, in der der Arbeiter abwesend war

Tabelle A.7: Parameter der Entität Arbeiter

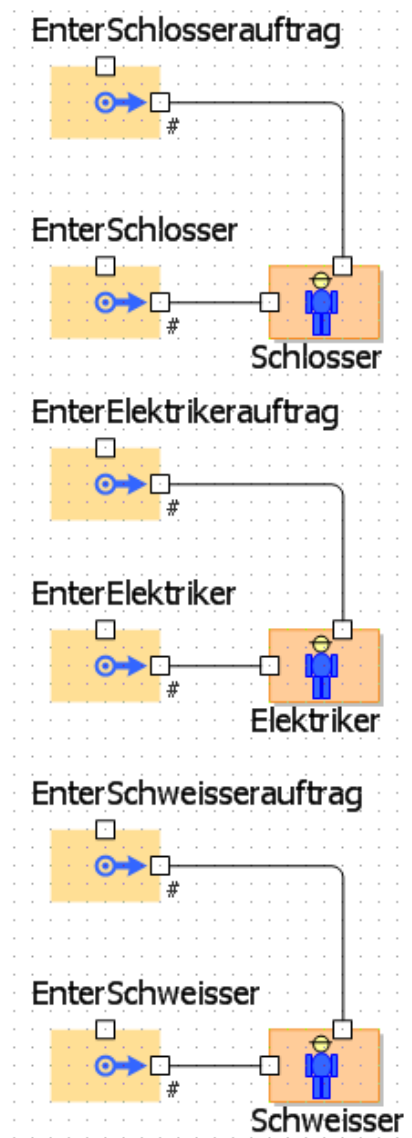


Abbildung A.2: Arbeiterobjekte im Modell

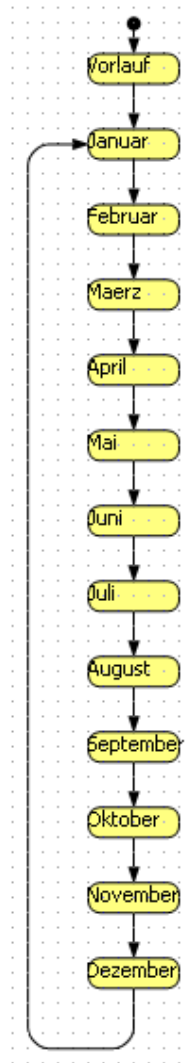


Abbildung A.3: Statechart zur Steuerung Wahrscheinlichkeit der Ausfallszeiten

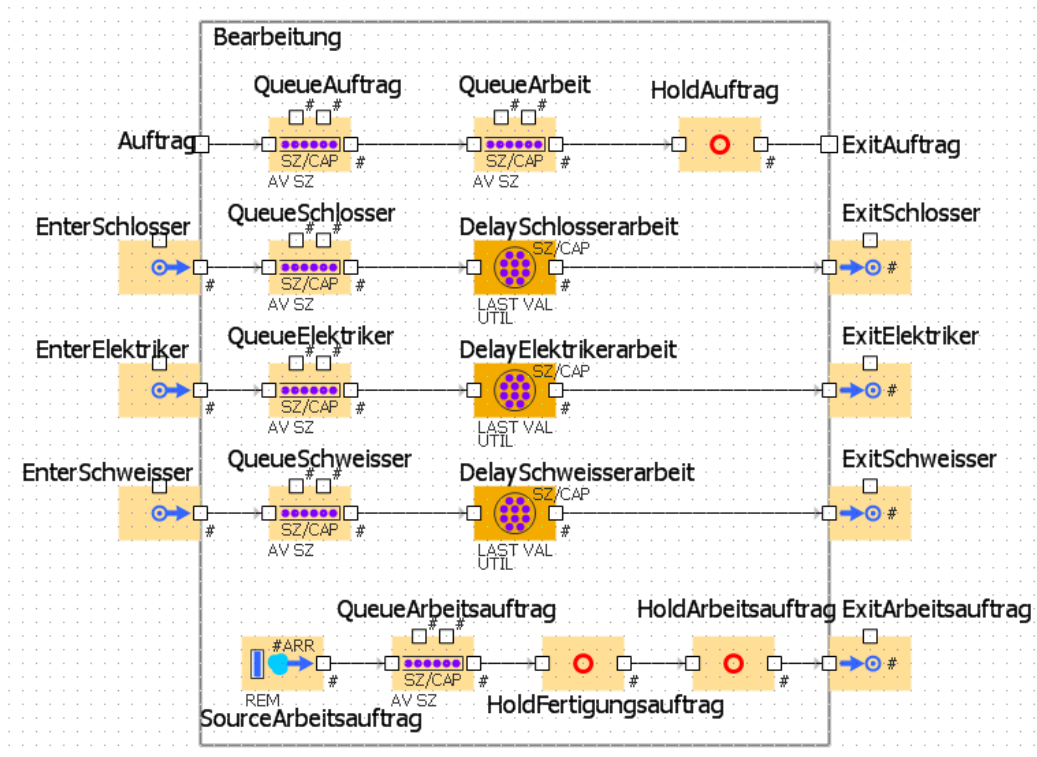


Abbildung A.4: Objekt Bearbeitung

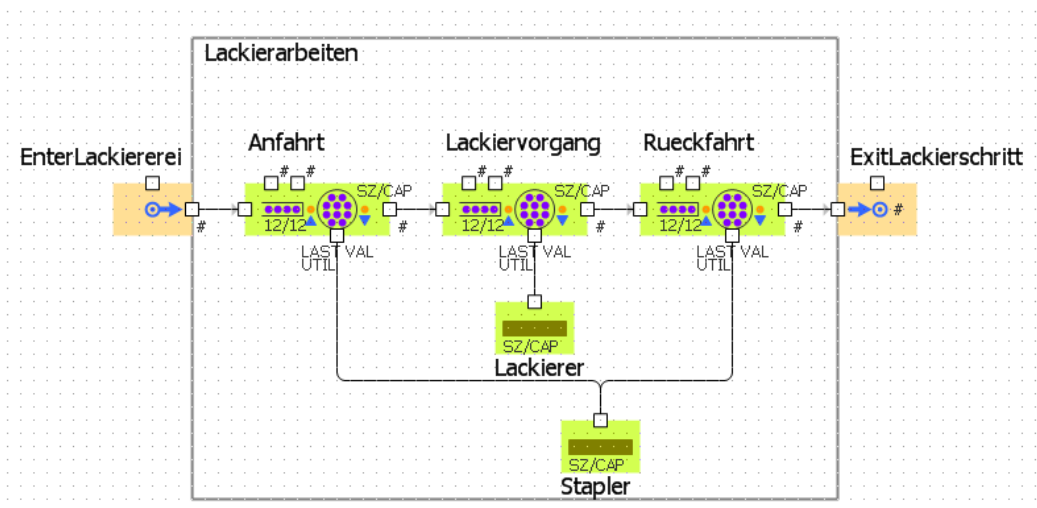


Abbildung A.5: Objekt Lackieren

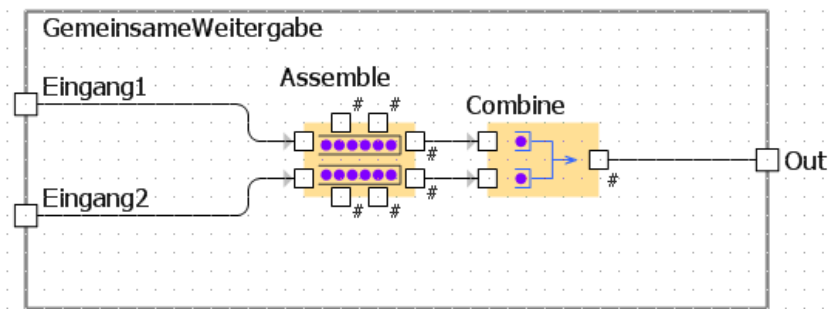


Abbildung A.6: Objekt Gemeinsame Weitergabe

Anhang B

Programmierung

Das Verhalten eines Objektes kann mit Einstellungsmöglichkeiten von AnyLogic und Parametern beeinflusst werden. Weiters stehen vordefinierte Objekte sowie Methoden zur Verfügung die direkt verwendet werden können. Dazu gehören Formeln, Gleichungen, Timer und Sequenzdiagramme. Sollten diese Einstellungsmöglichkeiten nicht mehr ausreichen, so kann mittels direkter Programmierung in Java das gewünschte Verhalten programmiert werden. Die Java-Methoden werden dann vom Objekt zu den jeweils definierten Aktionen aufgerufen. Im Anschluss werden einige relevante Methoden des Modells angegeben.

B.1 Allgemeine Methoden für das Modell

```
//Kundenauftrag zu entsprechender Lackierstation schicken
public void sendLackieren(Entity myEntity, String zielstation) {
    Kundenauftrag myAuftrag = (Kundenauftrag)myEntity;
    myAuftrag.lackierschritt = zielstation;
    Lackierarbeiten lackierarbeiten = (Lackierarbeiten)
    (Engine.getRoot()).findObjectByName("Lackiererei");
    lackierarbeiten.EnterLackiererei.take(myEntity);
}
```

```
// Neuen Auftrag in das System stellen
public void auftragAusschreiben(Entity myEntity) {
    Kundenauftrag myAuftrag = (Kundenauftrag)myEntity;
    myAuftrag.Update(Startzeitpunkt);
    //Fahrzeugtyp zufällig auswählen
    myAuftrag.fahrzeugtyp = (int)(AuswahlFahrzeugtyp(uniform()));
}
```

```

//Wenn normaler Roto: Classic- oder Standardverkleidung
zufällig auswählen
if (myAuftrag.fahrzeugtyp == 2) {
    myAuftrag.verkleidung = (int)(AuswahlVerkleidung(uniform()));
//bei ABRA-> Styling Verkleidung, bei Variopress nur ein
Verkleidungstyp
} else {
    myAuftrag.verkleidung = 2;
}
}

```

B.2 Methode für das Objekt "Arbeitersource"

```

//Arbeitsauftrag an Arbeiter übergeben und Arbeiter zur Arbeits-
station schicken
public void sendWorker(Entity myEntity) {
    Arbeiter myArbeiter = (Arbeiter)myEntity;
    String arbeitsstation = myArbeiter.arbeitsstation;
    String arbeiterart = myArbeiter.arbeiterart;
    Bearbeitung myBearbeitung =
    (Bearbeitung)(Engine.getRoot()).findObjectByName(arbeitsstation);
    Enter myEntry = null;
    if (arbeiterart == "Schlosser") {
        myEntry = (Enter)myBearbeitung.EnterSchlosser;
    } else {
        if (arbeiterart == "Elektriker") {
            myEntry = (Enter)myBearbeitung.EnterElektriker;
        } else {
            if (myBearbeitung != null) {
                myEntry = (Enter)myBearbeitung.EnterSchweisser;
            }
        }
    }
    myEntry.take(myEntity);
}

```

B.3 Methoden für das Objekt "Bearbeitung"

```
//Auswahl der Verteilungen der Bearbeitungszeiten aus der Datenbank
public void zeitenAnfragen(final int stationName , final int type,
    final int verkleidungstyp) {
    String feldVal = "xx/xx/xx";
    String []vals = feldVal.split("/");
    double minVal = 0;
    double peakVal = 0;
    double maxVal = 0;
    double vert = 0;
    double vert1 = 0;
    double vert2 = 0;
    double vert3 = 0;
    int vergleich = 0;
    //Abfrage für die Datenbank erstellen
    StringBuffer mySQL = new StringBuffer();
    mySQL.append("Select * FROM Modeldata WHERE Station = ");
    mySQL.append(stationName);
    mySQL.append(" AND Type = ");
    mySQL.append(type);
    mySQL.append(" AND Cover = ");
    mySQL.append(verkleidungstyp);
    try {
        //Datesatz einlesen
        ResultSet rs = TimeData.getResultSet(mySQL.toString());
        rs.next();
        // Schlosserzeiten herausuchen
        vergleich = rs.getInt("NoMechanic");
        if(vergleich >0) {
            vert = uniform();
            vert1 = rs.getDouble("MechanicProb1");
            vert2 = rs.getDouble("MechanicProb2");
            vert3 = rs.getDouble("MechanicProb3");
            if (vert <= vert1) {
                feldVal = rs.getString("MechanicDist1");
            } else {
                if (vert <= vert2) {
                    feldVal = rs.getString("MechanicDist2");
                } else {
                    feldVal = rs.getString("MechanicDist3");
                }
            }
        }
    }
}
```

```
}
}
vals = feldVal.split("/");
//Dreiecksverteilung für Schlosser festlegen
minVal = Double.valueOf(vals[0]).doubleValue();
peakVal = Double.valueOf(vals[1]).doubleValue();
maxVal = Double.valueOf(vals[2]).doubleValue();
delaySchlosser = triangular(minVal,peakVal,maxVal);
DelaySchlosserarbeit.unblock();
}

// Elektrikerzeiten herausuchen
vergleich = rs.getInt("NoElectrician");
if(vergleich >0) {
    vert = uniform();
    vert1 = rs.getDouble("ElectricianProb1");
    vert2 = rs.getDouble("ElectricianProb2");
    vert3 = rs.getDouble("ElectricianProb3");
    if (vert <= vert1) {
        feldVal = rs.getString("ElectricianDist1");
    } else {
        if (vert <= vert2) {
            feldVal = rs.getString("ElectricianDist2");
        } else {
            feldVal = rs.getString("ElectricianDist3");
        }
    }
}
vals = feldVal.split("/");
//Dreiecksverteilung für Elektriker festlegen
minVal = Double.valueOf(vals[0]).doubleValue();
peakVal = Double.valueOf(vals[1]).doubleValue();
maxVal = Double.valueOf(vals[2]).doubleValue();
delayElektriker = triangular(minVal,peakVal,maxVal);
DelayElektrikerarbeit.unblock();
}

// Schweisserzeiten herausuchen
vergleich = rs.getInt("NoWelder");
if(vergleich >0) {
    vert = uniform();
    vert1 = rs.getDouble("WelderProb1");
```



```

        vert2 = rs.getDouble("WelderProb2");
        vert3 = rs.getDouble("WelderProb3");
        if (vert <= vert1) {
            feldVal = rs.getString("WelderDist1");
        } else {
            if (vert <= vert2) {
                feldVal = rs.getString("WelderDist2");
            } else {
                feldVal = rs.getString("WelderDist3");
            }
        }
        vals = feldVal.split("/");
        //Dreiecksverteilung für Schweisser festlegen
        minVal = Double.valueOf(vals[0]).doubleValue();
        peakVal = Double.valueOf(vals[1]).doubleValue();
        maxVal = Double.valueOf(vals[2]).doubleValue();
        delaySchweisser = triangular(minVal,peakVal,maxVal);
        DelaySchweisserarbeit.unblock();
    }

} catch (java.sql.SQLException e) {
    System.out.println("Fehler: " + e.getMessage());
    traceln("Fehler bei Bearbeitung/ Zeiten: " + e.getMessage());
}
}

// Wenn Auftrag zur Bearbeitung wartet, Arbeitsaufraege erstellen
public void ressourcenAnfragen(final int stationName, final int type,
    final int verkleidungstyp) {
    int counter = 0;
    //Abfrage der benötigten Arbeiter für die Datenbank erstellen
    StringBuffer mySQL = new StringBuffer();
    mySQL.append("Select * FROM Modeldata WHERE Station = ");
    mySQL.append(stationName);
    mySQL.append(" AND Type = ");
    mySQL.append(type);
    mySQL.append(" AND Cover = ");
    mySQL.append(verkleidungstyp);
    try {
        //Datensatz einlesen

```

```
ResultSet rs = TimeData.getResultSet(mySQL.toString());
if (rs != null && rs.next()) {
    anzSchlosser = rs.getInt("NoMechanic");
    anzElektriker = rs.getInt("NoElectrician");
    anzSchweisser = rs.getInt("NoWelder");
    if (anzSchlosser > 0) {
        QueueSchlosser.unblock(); // Eingang für Schlosser öffnen
        for(int i=1;i <= anzSchlosser; i++) {
            // Arbeitsauftrag mit richtigen Werten versehen
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            arbeitsstation = this.stationsaufruf;
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            arbeiterart = "Schlosser";
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            priority = QueueAuftrag.size() + stationsnr;
            counter++;
        }
    }
    if (anzElektriker >0) {
        QueueElektriker.unblock(); // Eingang für Elektriker öffnen
        for(int i=1;i<=anzElektriker; i++) {
            // Arbeitsauftrag mit richtigen Werten versehen
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            arbeitsstation = this.stationsaufruf;
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            arbeiterart = "Elektriker";
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            priority = QueueAuftrag.size() + stationsnr;
            counter++;
        }
    }
    if (anzSchweisser >0) {
        QueueSchweisser.unblock(); // Eingang für Schweisser öffnen
        for(int i=1;i <= anzSchweisser; i++) {
            // Arbeitsauftrag mit richtigen Werten versehen
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            arbeitsstation = this.stationsaufruf;
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            arbeiterart = "Schweisser";
            ((Arbeitsauftrag)(QueueArbeitsauftrag.get(counter))).
            priority = QueueAuftrag.size()+ stationsnr;
        }
    }
}
```

```

        counter++;
    }
}
//Arbeitsauftragsentities sind bereit um Arbeiter abzuholen
HoldArbeitsauftrag.unblock();
}
} catch (java.sql.SQLException e) {
    System.out.println("Fehler: " + e.getMessage());
    traeln("Fehler bei Bearbeitung/ Ressourcen: " + e.getMessage());
}
}
}

```

B.4 Methode für das Objekt "Lackierarbeiten"

```

//Lackierzeiten bei jedem Lackierschritt für die Auftragsvariante
aus Datenbank einlesen
public void zeitenAnfragen(final String lackierschritt,
    final int type, final int verkleidungstyp) {
    String getField = new String();
    double getVert = 0;
    int getWert = 0;
    int stationName = 0;
    String feldVal = "xx/xx/xx";
    String []vals = feldVal.split("/");
    double minVal = 0;
    double peakVal = 0;
    double maxVal = 0;

    double vert = 0;
    double vert1 = 0;
    double vert2 = 0;
    double vert3 = 0;

    //Abhängig von der Lackierstation, in der sich der Auftrag
    befindet, die Zeiten raussuchen
    //Zuerst richtige Stationsnummer je Lackierschritt festlegen
    if (lackierschritt == "EnterLackierschritt1") {

```

```
        stationName = 3;
    }
    if (lackierschritt == "EnterLackierschritt2") {
        stationName = 8;
    }
    if (lackierschritt == "EnterLackierschritt3") {
        stationName = 14;
    }
    if (lackierschritt == "EnterLackierschritt4") {
        stationName = 28;
    }
    if (lackierschritt == "EnterLackierschritt5") {
        stationName = 31;
    }
    if (lackierschritt == "EnterLackierschritt6") {
        stationName = 17;
    }
    //Abfrage der Zeiten aus Datenbank ausführen
    StringBuffer mySQL = new StringBuffer();
    mySQL.append("Select * FROM Modeldata WHERE Station = ");
    mySQL.append(stationName);
    mySQL.append(" AND Type = ");
    mySQL.append(type);
    mySQL.append(" AND Cover = ");
    mySQL.append(verkleidungstyp);

    try {
        ResultSet rs = TimeData.getResultSet(mySQL.toString());
        rs.next();
        //Montagezeiten entsprechend der Verteilungen heraussuchen
        vert = uniform();
        vert1 = rs.getDouble("PainterProb1");
        vert2 = rs.getDouble("PainterProb2");
        vert3 = rs.getDouble("PainterProb3");
        if (vert <= vert1) {
            feldVal = rs.getString("PainterDist1");
        } else {
            if (vert <= vert2) {
                feldVal = rs.getString("PainterDist2");
            } else {
                feldVal = rs.getString("PainterDist3");
            }
        }
    }
}
```

```
    }  
    }  
    vals = feldVal.split("/");  
//Richtige Werte (Minimum/ Peak/ Maximum) für Dreiecksverteilung  
    minVal = Double.valueOf(vals[0]).doubleValue();  
    peakVal = Double.valueOf(vals[1]).doubleValue();  
    maxVal = Double.valueOf(vals[2]).doubleValue();  
    delayLackierer = triangular(minVal,peakVal,maxVal);  
} catch (java.sql.SQLException e) {  
    traceln("Fehler beim Lackieren" + e.getMessage());  
    traceln("Station: " + stationName + "/" Typ: " + type +  
        "/Verkleidung: " + verkleidungstyp);  
    System.out.println("Fehler: " + e.getMessage());  
}  
}
```

Anhang C

Kurzfassung

Diese Arbeit behandelt das Problem der variablen Fließfertigung mit frei beweglichen Arbeitern unter stochastischen Bedingungen. Als Praxisbeispiel dient dabei die Fahrzeugmontage der Firma M-U-T. Diese geschieht derzeit in Einzelplatzmontage und in Fertigungsinseln, soll aber auf eine Fließfertigung umgestellt werden.

Als Basis für die Auslegung eines Fließfertigungssystems soll ein Simulationsmodell dienen welches in dieser Arbeit formuliert, erstellt und untersucht wird. Das Modell hat zum Ziel, einen Fertigungshorizont von einem Jahr darstellen können. Die Auftragsgenerierung sowie die Bearbeitungszeiten in den einzelnen Stationen ist stochastisch. Weiters sind die Arbeiter zwischen den Stationen frei beweglich. Insgesamt sind nicht genügend Arbeiter verfügbar um alle Stationen gleichzeitig zu bedienen. Das Modell berücksichtigt auch den Ausfall von Arbeitern auf Grund von Urlaub oder Krankenstand.

Die Aufgabenstellung mit der Ausgangssituation wird dabei in Kapitel 2 genau dargestellt. Hier wird die Problemstellung der Firma M-U-T, welche als Praxisbeispiel dient, genau erläutert. Weiters wird auf Arbeiten anderer Autoren verwiesen die ähnliche Aufgabenstellungen behandelt haben. Es werden auch Hinweise zu weiterer Literatur gegeben die nicht in unmittelbarem Zusammenhang zu dieser Arbeit stehen aber durchaus interessante Randbereiche behandeln. Damit wird ein Überblick über dieses Forschungsfeld gegeben. Im Anschluss daran werden im Kapitel 3 Struktur- und Planungsmodelle für Produktionssysteme gezeigt. Geeignete Modelle für diese Aufgabe werden diskutiert und verglichen um schließlich das Fertigungsmodell der M-U-T zu kategorisieren. Im Kapitel 4 wird auf die verschiedenen Möglichkeiten der Optimierung und mathematischen Problemlösung mittels Operations Research eingegangen. Insbesondere werden die Bereiche Lineare Optimierung, Netzplantechnik, Warteschlangentheorie und Simulation genauer behandelt und gegenübergestellt.

Die Umsetzung des Simulationsmodelles ist in Kapitel 5 dargestellt. Nach der Formulierung der Modellannahmen sowie der Datensammlung und -auswertung wird die Layouterstellung des Simulationsmodells in AnyLogic behandelt. Nach der Validierung und Verifizierung des Modells kann schließlich in Kapitel 6 mit den Untersuchungen und Versuchen am Modell begonnen werden. In diesem Kapitel wird das Experimentdesign sowie die Durchführung dargestellt und die Ergebnisse präsentiert und diskutiert. Die Zusammenfassung der Ergebnisse, eine kritische Betrachtung sowie ein Ausblick wird abschließend in Kapitel 7 gegeben. Im Anhang werden die Objekte des AnyLogic- Modells gezeigt und ausgewählte Java- Methoden angegeben.

Anhang D

Summary

This work deals with the problem of variable flow-shop production with moveable workers under stochastic conditions. The vehicle assembly of the company M-U-T serves as practical example. The assembly is now conducted on single assembly stations and manufacturing cells. This system should be changed to a flow-shop model.

To provide a basis for the design of a flow-shop system a simulation model has to be created, tested and investigated in this work. The model should be able to show a production cycle of one year. Generation of orders and working times in the stations are stochastic. Furthermore the workers can be moved freely between the stations. In total there are not enough workers available to operate all stations at the same time. The model also considers failure of workers because of vacation or illness.

The problem definition and current situation is shown in detail in chapter 2. The task of the company M-U-T, which serves as an example, is explained in detail. There are also references to papers of other authors who dealt with similar problems. Additionally there are references to literature which has no direct correlation to the current work but deal with familiar areas of research. This provides an overview of this field of research. Subsequent chapter 3 shows some structural and planning models for production systems. Appropriate models for this work are discussed and compared in order to categorize the production system of M-U-T. Chapter 4 afterwards shows several opportunities for optimization and mathematical model solving with operations research. Especially the linear optimization, network techniques, queuing theory and simulation is dealt and compared in detail.

The conversion of the simulation model is shown in chapter 5. After formulation of model assumption, data collection and analysis the model layout design with AnyLogic is explained. Following the verification and validation procedures chapter 6 then starts with the model analysis and model experiments.

This chapter explains the model design and performance and discusses the simulation results. Conclusions, critical model review and an outlook is given in chapter 7.

The annex finally shows the actual AnyLogic models and some important Java-methods within the model.

Anhang E

Lebenslauf



Persönliche Daten

Vorname / Name: Martin Keiml
 Geburtsdatum: 27. Dezember 1979
 Geburtsort: Wien/ Österreich
 Staatsangehörigkeit: Österreich

Ausbildung

10.2000 – 11.2008

Betriebswirtschaftliches Zentrum Hauptuniversität Wien, Wien

Fachrichtung Internationale Betriebswirtschaft
 Schwerpunkte/ KFK: Operations Research,
 Produktionsmanagement, Controlling
 Diplomarbeit: Simulationsmodell einer variablen
 Kleinserienfertigung an Hand eines praktischen Beispiels
 Abschluß: Mag. rer. soc. oec, Notenschnitt: 2,3

1994 – 1999

HTBLA Hollabrunn, Hollabrunn

Fachrichtung Wirtschaftsingenieurwesen
 Schwerpunkte: Qualitätsmanagement, Betriebsführung
 Ingenieursprojekt: Multivariate Sorptionsuntersuchung an
 Deponiewerkstoffen
 Abschluß: Matura; Gesamtnote: 1,2

Auslandsaufenthalte

09/2003 – 08/2004

Auslandsstudium an der Universität von Kyoto, Japan:
 Schwerpunkte waren Sprachkurse Japanisch, japanische
 Landeskunde sowie japanische Wirtschaft

02/2005

Zweiwöchiger Sprachaufenthalt in Moskau, Russland:
 Intensivsprachkurs der von der Universität Wien organisiert
 wurde

Beruflicher Werdegang

10/2008 – jetzt

ASFINAG Baumanagement GmbH, Wien

Planung und Durchführung von Bauprojekten am hochrangigen Straßennetz in Österreich

1 Mrd. € Umsatz, ca. 230 Mitarbeiter

Qualitätsmanager

- Leitung des Prozess- und Qualitätsmanagement
- Aktualisierung von Prozessen
- Organisation und Durchführung interner und externer Audits, Prozess- Reviews und Schulungen
- Erstellung und Monitoring einer Balanced Scorecard
- Abstimmung mit dem ASFINAG Konzern bzgl. Prozess- und Qualitätsmanagement

08/2000 – 09/2008

M-U-T Maschinen-Umwelttechnik-Transportanlagen GmbH, Stockerau

Maschinen- und Anlagenbau, Kommunalfahrzeugbau; Hersteller von Abfallsammelfahrzeugen, Kehrmaschinen, Kanalreinigungsfahrzeugen, Kläranlagen, Kompostieranlagen, Förder- und Sortieranlagen

32 Mio. € Umsatz; ca. 170 Mitarbeiter

Konstrukteur

- Konstruktion von Förderanlagen
- Überarbeitung und Neukonstruktion von Fahrzeugkomponenten
- Wertanalytische Betrachtung von Abfallsammelfahrzeugen, Einführung eines Baukastensystems zur Kleinserienfertigung

Mitarbeiter Qualitätsmanagement (seit 2002)

- Erste Arbeiten im Qualitätsmanagement (Überarbeitung von Verfahrens- und Arbeitsanweisungen und Dokumenten)
- Projektleiter- Stv. beim Umstieg auf die DIN EN ISO 9001:2000
- Reorganisation der Qualitätssicherung zum modernen Qualitätsmanagement
- Einführung und Programmierung diverser MS Access-Datenbanken zur Unterstützung von Prozessen

Qualitäts-, Sicherheits- und Umweltmanagementbeauftragter (seit 03/2006)

- Leitung interner Audits für Qualitätsmanagement und Arbeitssicherheit
- Lenkung und Überarbeitung der Dokumente und Aufzeichnungen sowie Prozesse
- Erstellung von Lösemittelbilanzen und ARA- Meldungen
- Erstellung des Abfallwirtschaftskonzepts
- Beauftragter für die Arbeitssicherheit im Unternehmen und auf Baustellen
- Überwachung der Prozesse und Abläufe auf Einhaltung der Vorgaben der DIN EN ISO 9001:2000 und SCC

Weiterbildung / Qualifikationen

01/2006	Qualitätsmanagement-Beauftragter für KMU, WIFI Österreich, Wien
02/2006	Qualitätsmanagement-Auditor, WIFI Österreich, Wien
06/2006	SCC- Schulung für operative Mitarbeiter
01/2007	Controlling mit SAP R/3
06/2007	Ausbildung zum Abfallbeauftragten, Wirtschaftsuniversität, Wien

Besondere Kenntnisse

Fremdsprachen

Englisch: fließend in Wort und Schrift
 Japanisch in Wort (gut) und Schrift (Grundkenntnisse)
 Russisch: Grundkenntnisse in Wort und Schrift

EDV-Kenntnisse

Betriebssysteme	MS Windows
Betriebswirtschaft	Infor BRAIN, SAP/ R3
Datenbank	MS Access, MS SQL 2000
Qualitätsmanagement	ADONIS- ADOquality

Stockerau, 15.11.2008

Ort, Datum

Kem

Unterschrift