

Modellierung und Bewertung von Fertigungssystemen mit Petri-Netzen

vorgelegt von
Diplom-Informatiker
Armin Zimmermann

Vom Fachbereich 13 - Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor-Ingenieur
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr.-Ing. Peter Pepper
Berichter: Prof. Dr.-Ing. Günter Hommel
Berichter: Prof. Dr.-Ing. Eckehard Schnieder
Berichter: Prof. Dr.-Ing. Adam Wolisz

Tag der wissenschaftlichen Aussprache: 3. September 1997

Berlin 1997

D 83

Zusammenfassung

Bei der Planung moderner Fertigungssysteme ist der Einsatz rechnergestützter Modellierungsmethoden notwendig, mit denen Verhalten und Leistungsfähigkeit eines Fertigungsprozesses untersucht und bewertet werden können. Ziel dieser Dissertation ist die Unterstützung des Entwurfs von Fertigungssystemen mit einer auf Petri-Netzen basierenden Modellierungs- und Analysemethodik. Petri-Netze erlauben die Beschreibung von diskreten Systemen, die durch Nebenläufigkeit, Synchronisation, Konflikte und zeitverbrauchende Aktivitäten gekennzeichnet sind.

Für die einfache und komfortable Abbildung von Fertigungsprozessen wurde eine spezielle Art farbiger Petri-Netze entwickelt, die für das Anwendungsgebiet angepaßt ist. Damit ist die getrennte Modellierung von Struktur und Arbeitsplänen eines Fertigungssystems mit derselben Beschreibungsmethode möglich. Das Strukturmodell entspricht dabei weitgehend dem realen Aufbau des Fertigungssystems, was die Verständlichkeit erleichtert und eine spätere Visualisierung des Verhalten ermöglicht. Um die Entwicklung großer Modelle zu erleichtern, wird eine Bibliothek von Teilmodellen eingesetzt. Ein solches Modell steht für eine Gruppe ähnlicher Fertigungsstrukturen, die durch unterschiedliche Parametrisierung aus dem Grundmodell abgeleitet werden. Aus den die Struktur und die Arbeitspläne beschreibenden Modellteilen wird automatisch ein Gesamtmodell erzeugt. Dazu wird das Strukturmodell um Informationen aus den Arbeitsplänen angereichert.

Anhand des so erzeugten Modells werden dann Eigenschaften des Fertigungssystems bestimmt. Qualitative Aussagen geben Aufschluß über die korrekte Funktionsweise des modellierten Systems. Durch numerische Analyse oder Simulation können die Werte von Leistungsparametern bestimmt und damit die Leistungsfähigkeit und Zuverlässigkeit eines geplanten Systems überprüft werden. Mit der Vorhersage von Eigenschaften eines Fertigungssystems kann die Planung sicherer stattfinden. Durch den Vergleich verschiedener Varianten eines Fertigungsprozesses anhand ihrer Leistungsfähigkeit ist eine Optimierung möglich. Alternativ zur Modellierung mit speziellen Petri-Netzen wird die Beschreibung eines Fertigungssystems mit Funktionssymbolen angeboten. Damit ist die Verwendung der Analysetechniken auch ohne Kenntnis von Petri-Netzen möglich. Ein solcherart spezifiziertes Modell wird automatisch in ein spezielles Petri-Netz übersetzt, um danach untersucht werden zu können.

Die beschriebenen Methoden wurden in einem Softwarewerkzeug realisiert. Aufgrund der verschiedenartigen Beschreibungsmittel ist dafür eine parametrisierbare Benutzungsoberfläche für hierarchische, grafische Modelle implementiert worden. Mit Hilfe dieses Werkzeugs wurden die entwickelten Techniken an einem realen Anwendungsbeispiel auf ihre Verwendbarkeit überprüft.

Abstract

The design of a manufacturing system requires modeling and performance evaluation techniques. A modeling and analysis method based on Petri nets is proposed in this thesis, that aims at supporting the design of manufacturing systems. Petri nets allow the representation of discrete systems containing concurrent, synchronized, conflicting, and timed activities.

A class of dedicated coloured Petri nets is introduced, which is especially tailored to manufacturing systems and facilitates their convenient modeling. The structure and the work plans of a manufacturing system can both be modeled separately using the same method. The structural model resembles the layout of the modeled system. This is advantageous for its understandability and a later visualization of the behavior. A library of model templates helps to create large models. Each template represents a set of similar manufacturing cells or other structures, that can be derived from the template by different parameter assignments. The structure and production route models are automatically merged to create a complete model of the manufacturing system. During this process, the information contained in the work plan models is added to the structural model.

Measures of interest can be obtained from the complete model. The correct behavior of the modeled system can be checked by qualitative analysis techniques. Quantitative properties are derived by numerical analysis or simulation, showing the performance and dependability of the considered system. The design process becomes more reliable with the opportunity to calculate system properties. An optimization is possible by evaluating and comparing measures of interest for different model variations. A modelling method based on functional blocks can be used alternatively. The analysis techniques can thus be used without knowledge of Petri nets. Such a symbolic model is automatically translated into a dedicated Petri net and can be evaluated afterwards.

The mentioned modeling and analysis techniques are implemented in a software tool. A generic graphical user interface for a wide range of hierarchical models has been developed for the different modeling methods. The application of the proposed modeling and analysis techniques to a real-life manufacturing system, carried out using the software tool, shows the usefulness of this approach.

Danksagung

Mein Dank gilt allen, die mich während der Vorarbeiten und der Fertigstellung dieser Dissertation fachlich und persönlich unterstützt haben. Sehr wichtig waren mir die Anregungen und Hinweise von Prof. Günter Hommel, meinem Betreuer während der vergangenen drei Jahre. Das Klima an seinem Lehrstuhl empfand ich als angenehm und fachlich anregend, wofür natürlich auch meinen Kollegen zu danken ist. Unter ihnen möchte ich meinen besonderen Dank Reinhard German und Christian Kelling aussprechen, in deren Forschungsarbeiten wichtige Voraussetzungen für die vorliegende Arbeit entstanden.

Der Deutschen Forschungsgemeinschaft danke ich für die Möglichkeit, innerhalb des Graduiertenkollegs „Kommunikationsbasierte Systeme“ und eines Forschungsprojektes im Schwerpunktprogramm „Modellierung der Produktion“ die hier vorgestellten Forschungsarbeiten durchführen zu können. Die Diskussionen mit Professoren und Kollegiaten im Graduiertenkolleg sowie Fragen von Gutachtern und anderen Teilnehmern des Schwerpunktprogramms ergaben wichtige Impulse für die Arbeit.

Für die engagierte und ideenreiche Mitarbeit an der Realisierung der hier vorgestellten Methoden danke ich den Studenten Frank Bayer, Mario Beck, Stefan Bode, Knut Dalkowski, Kolja Koischwitz und Andreas Kühnel. Mit ihren Diplomarbeiten und während ihrer Projektstätigkeit hatten sie entscheidenden Anteil an der Umsetzung der Forschungsergebnisse.

Ein besonderer Dank gilt meiner Frau Anja, ohne deren Unterstützung und Rücksicht besonders während der letzten Monate die Fertigstellung dieser Arbeit kaum möglich gewesen wäre.

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	ii
Danksagung	iii
1 Einleitung	1
1.1 Motivation	1
1.2 Petri-Netze	2
1.3 Aufgabenstellung und Beitrag	4
1.4 Aufbau der Arbeit	6
2 Modellierung von Fertigungssystemen	8
2.1 Anforderungen und Problemstellungen	8
2.2 Ein Beispiel-Fertigungssystem	11
2.3 Klassische Stochastische Petri-Netze	15
2.4 Farbige Petri-Netze	18
2.5 Andere Beschreibungsmittel	21
2.6 Einordnung der Arbeit	23
3 Modellierung von Fertigungssystemen mit speziellen Petri-Netzen	26
3.1 Eine Petri-Netz-Klasse für Fertigungssysteme	27
3.1.1 Definition	27
3.1.2 Erläuterungen	32
3.2 Getrennte Modellierung von Fertigungssystem und Arbeitsplänen	38
3.3 Eine Bibliothek von Teilmodellen	51
3.4 Automatische Erzeugung eines Gesamtmodells	60

4	Bewertungsverfahren	66
4.1	Fragestellungen und Leistungsmaße	66
4.2	Qualitative Eigenschaften und ihre Bestimmung	68
4.3	Quantitative Eigenschaften	75
4.3.1	Untersuchung des Netzverhaltens	76
4.3.2	Numerische Analyse	86
4.3.3	Simulation	90
5	Funktionssymbole als alternatives Beschreibungsmittel	94
5.1	Verwendung der Modulbibliothek	94
5.2	Symbolische Modellierung von Fertigungssystemen	97
5.3	Umwandlung in ein spezielles Petri-Netz	100
6	Ein Anwendungsbeispiel	104
6.1	Beschreibung des Fertigungssystems	104
6.2	Modellierung mit speziellen Petri-Netzen	111
6.3	Untersuchung und Bewertung	119
6.4	Fazit	129
7	Ein Softwarewerkzeug zur Modellierung und Bewertung von Fertigungssystemen mit speziellen Petri-Netzen	131
7.1	Eine generische Benutzungsoberfläche	132
7.2	Werkzeugkomponenten zur Modellbewertung	135
7.3	Zusammenfassung	138
8	Zusammenfassung und Ausblick	139
Anhang		
A	Symbole und Abkürzungen	142
B	Multimengen	144
C	Übergangstabellen und Ausdrücke	145
D	Formate von Modellbeschreibungen	147
	Literaturverzeichnis	150

Kapitel 1

Einleitung

Durch den technischen und wissenschaftlichen Fortschritt werden immer komplexere Techniken, Verfahren und Maschinen entwickelt und eingesetzt. Die erforderlichen Anlaufinvestitionen steigen, bevor ein Nutzen zu erzielen ist. Dem Entwicklungs- bzw. Entwurfsprozeß kommt demnach eine hohe Bedeutung zu. Dabei sind Aussagen über das spätere Verhalten eines geplanten Systems für fundierte Entscheidungen notwendig. Neben Fragen nach der korrekten Funktionsweise sind besonders Aussagen über die Leistungsfähigkeit und Zuverlässigkeit interessant. Ein wichtiges Hilfsmittel zur Lösung der genannten Probleme stellt die Modellierung und Untersuchung der geplanten Systeme dar. Ein Modell kann außerdem für den Informationsaustausch zwischen den Entwurfsebenen eines Fertigungssystems verwendet werden.

Thema dieser Dissertation ist eine Modellierungs- und Bewertungsmethode für Fertigungssysteme. Dafür werden spezielle zeiterweiterte farbige Petri-Netze verwendet, einem aus der Informatik stammenden grafischen Modellierungsmittel. Diskrete Fertigungssysteme sind durch beschränkte Ressourcen, Konflikte, Nebenläufigkeit und Synchronisation gekennzeichnet, die mit ihnen gut abgebildet werden können.

1.1 Motivation

Die oben genannten Probleme gelten auch und insbesondere für den Entwurf von Fertigungssystemen. Zu aktuellen Entwicklungstendenzen in der Fabrikplanung zählen die wachsende Variantenvielfalt der Produkte, der zunehmende Automatisierungsgrad der Anlagen, kurzfristige und kundenspezifische Aufträge, eine höhere Integration von Material- und Informationsfluß sowie eine stärkere Differenzierung der Organisationstypen des Fabrikbetriebes [81, 66]. Es werden demnach anpassungsfähige Produktionsstrukturen benötigt, die auch für kleine Losgrößen wirtschaftlich sind. Die eingesetzten Verfahren der Produktionsplanung und -steuerung müssen zu geringer Durchlaufzeit und niedrigen Beständen führen. Eine schnelle und genaue Planung der Fabrik ist wichtig, denn die Phase der Systemplanung entscheidet über die spätere Funktionalität und Wirtschaftlichkeit. Die Unterstützung des Entwurfsprozesses durch Modellbildung und Untersuchung des geplanten Fertigungssystems ist darum notwendig. Die Modellierung erfordert ein geeignetes Beschreibungsmittel. Da-

bei ist es wichtig, sich auf eine für alle am Entwurf Beteiligten verbindliche und eindeutige Darstellungsform zu einigen, um Verständigungsprobleme zu verhindern.

Petri-Netze sind als mathematisch fundiertes und grafisches Modellierungsmittel prinzipiell gut dazu geeignet. Sie können auf den verschiedenen Ebenen von der Produktionsplanung bis hin zur Fertigungssteuerung eingesetzt werden. Im Gegensatz zu Simulationssprachen sind Petri-Netze aufgrund ihrer mathematischen Definition zugänglich für Analysemethoden. Bei der Simulation kann nicht garantiert werden, daß alle Zustände des modellierten Systems durchsucht wurden, so daß keine Aussagen über die korrekte Funktionsweise (z.B. Verklemmungsfreiheit) gemacht werden können. Trotz des hohen Zeitaufwands für Simulationen bleiben diese aber die einzige Bewertungsmöglichkeit, wenn das Modell eine Analyse nicht zuläßt.

Eine Methodik zur Modellierung und Bewertung von Fertigungssystemen erfordert neben einem Beschreibungsmittel eine Entwurfs- und Analysemethode sowie ein entsprechendes Werkzeug [63]. Für ersteres wird in der vorliegenden Arbeit eine für Fertigungssysteme angepaßte Art farbiger, zeiterweiterter Petri-Netze vorgeschlagen. Es wird dargestellt, wie damit Struktur und Funktion von Fertigungssystemen abgebildet werden können. Untersuchungsverfahren für die entstehenden Modelle werden vorgestellt, die die Beantwortung qualitativer und quantitativer Fragestellungen ermöglichen. In einem Softwarewerkzeug wurden diese Methoden realisiert.

1.2 Petri-Netze

Petri-Netze wurden von Carl Adam Petri [61] zur Modellierung nebenläufiger diskreter Systeme vorgeschlagen. Ein Petri-Netz läßt sich als gerichteter Graph darstellen, in dem Kanten die als Stellen und Transitionen bezeichneten Knoten verbinden. Stellen werden als Kreis gezeichnet und repräsentieren passive Elemente (Lager, Warteschlange), während die als Rechteck dargestellten Transitionen Aktivitäten (Fertigungsschritt, Ausfall einer Maschine) modellieren. Petri-Netze sind gut für die Beschreibung von Systemen geeignet, in denen konkurrierende, synchrone, verteilte und parallele Vorgänge auftreten [59]. Sie vereinen die Beschreibung von Aktivitäten und Zuständen eines Systems. Auch komplexe Systeme mit relativ großem Zustandsraum lassen sich mit ihnen übersichtlich und korrekt wiedergeben. Durch die lokale Beschreibung von Aktivitäten und ihres Einflusses auf den Zustand sind sie für Verfeinerung und modulare Konstruktion eines Modells geeignet.

Der Zustand des modellierten Systems wird durch die Markierung des Netzes, der Belegung der Stellen mit Marken, beschrieben. Das Petri-Netz kann dann als ein Systemmodell interpretiert werden, in dem in einer Markierung eine gewisse Zeit vergehen muß, bevor eine Aktivität stattfindet. Da die Aktivitäten durch das Schalten von Transitionen modelliert werden, ist die Zuordnung einer Schaltzeit zu den Transitionen der natürlichste Weg, zeitbehaftete Petri-Netze einzuführen. Zeitliche Aspekte können aber genauso auch mit Stellen verknüpft werden. Mit Hilfe dieser Erweiterung ist es möglich, Leistungskenngrößen von Modellen wie z.B. Durchsatz, Auslastung oder Verfügbarkeit mit numerischen Methoden oder durch Simulation zu bestimmen.

Unter verschiedenen Vorschlägen zur Einbeziehung von Zeit in ein Petri-Netz-Modell sind besonders die stochastischen Petri-Netze (*stochastic Petri nets*, SPN) hervorzuheben, da

dieser Formalismus sich gut für die Analyse und Leistungsbewertung eignet [56, 2]. Dabei werden die Schaltzeiten der Transitionen als exponentiell verteilt angenommen. Der Erreichbarkeitsgraph eines solchen Modells wird auf eine zeitkontinuierliche Markow-Kette abgebildet, aus deren Lösung Aussagen über Leistungsparameter des Modells abgeleitet werden können. In anderen Arbeiten werden stochastische Petri-Netze mit diskreten Zeitabläufen untersucht [57], die sich auf diskrete Markow-Ketten abbilden lassen.

Um das Problem des oft sehr großen Zustandsraums und der entsprechend komplexen Lösung von SPNs anzugehen, wurden *generalized stochastic Petri nets* (GSPN) entwickelt, in denen Transitionen auch ohne Zeitverzug schalten können [4, 1, 16]. Dadurch enthält der Erreichbarkeitsgraph viele Markierungen, in denen der zugehörige stochastische Prozeß keine Zeit verbringt. Diese sogenannten verschwindenden Zustände haben keinen Einfluß auf die Komplexität der Lösung. Ebenso wie bei SPNs kann das Verhalten eines GSPNs über die Abbildung auf eine zeitkontinuierliche Markow-Kette untersucht werden.

Verschiedene Autoren beschäftigten sich mit der Erweiterung der Modellierungsmächtigkeit von Petri-Netzen, um auch nichtexponentielle Schaltzeiten verwenden zu können. Dabei steht die Entwicklung von numerisch handhabbaren Analysemethoden im Vordergrund, die die Berechnung von Leistungsgrößen für die entwickelten Modellklassen ermöglichen. Zusätzlich zu den zeitlosen und exponentiellen Transitionen können in *deterministic and stochastic Petri nets* (DSPN) Transitionen eine konstante Schaltzeit haben [5]. Wenn in keiner Markierung mehr als eine deterministische Transition schaltfähig ist, kann der stochastische Prozeß über eine eingebettete Markow-Kette untersucht werden.

In [32] werden *extended deterministic and stochastic Petri nets* (eDSPN) vorgeschlagen, in denen in jeder Markierung maximal eine Transition mit nichtexponentieller Schaltzeit aktiviert sein kann. Die nichtexponentiellen Schaltzeitverteilungen sind dabei kaum eingeschränkt, und gewünschte Verteilungen können approximiert werden. Die numerische Analyse des Verhaltens eines solchen Modells erfolgt über die Untersuchung des zugeordneten semi-regenerativen stochastischen Prozesses. Diese für farblose Petri-Netze vorgeschlagene Erweiterung wird in der vorliegenden Arbeit für die speziellen farbigen Petri-Netze und den ihr Verhalten beschreibenden stochastischen Prozeß angewandt. Weitere Erläuterungen dazu gibt Abschnitt 4.3.2. Eine Übersicht über die sich aus verschiedenen Modellarten ergebenden stochastischen Prozesse und Lösungsmethoden kann [20] entnommen werden.

Um komplexe Systeme auf einem höheren Abstraktionsniveau kompakter darstellen zu können, wurden farbige Petri-Netze [39, 40] und Prädikat-Transitions-Netze [28] vorgeschlagen. Einen Überblick über die Theorie und Anwendungen dieser sogenannten *high-level Petri nets* enthält [38]. Farbige Petri-Netze mit stochastischer Schaltzeitverteilung der Transitionen wurden von Zenie [82] sowie Lin und Marinescu vorgeschlagen [47, 48].

Aufgrund der Unteilbarkeit des Schaltens von Transitionen, der durch die möglichen Farben vorgegebenen endlichen Menge von Markenarten und der diskreten Marken sind Petri-Netze in der hier verwendeten Form für diskrete Systeme, nicht aber für kontinuierliche Modelle mit allmählichen Zustandsänderungen anwendbar. Für die Modellierung von Fertigungssystemen mit nicht zu vernachlässigenden kontinuierlichen Aspekten müssen andere Beschreibungsmittel verwendet werden, wie z.B. *fluid stochastic Petri nets* [74].

1.3 Aufgabenstellung und Beitrag

Diese Arbeit ist durch die zu Beginn der Einleitung genannten Probleme motiviert, die während des Entwurfs und der Änderung eines Fertigungssystems auftreten. Ziel ist eine durchgängige Modellierungs- und Analyseverfahren, die diesen Prozeß unterstützt. Insbesondere mit Hilfe der Leistungs- und Zuverlässigkeitsbewertung der entstehenden Modelle soll eine schnelle und sichere Vorhersage von Parametern des geplanten Systems möglich sein. Verschiedene Varianten können dann miteinander verglichen und die beste ausgewählt werden [67]. Damit ist indirekt eine Optimierung möglich, die ohne Modellierung und Bewertung nicht oder nur mit sehr hohem Aufwand erreichbar wäre. Im Abschnitt 2.1 werden die sich aus dieser Zielstellung ergebenden einzelnen Anforderungen an eine Modellierungs- und Analyseverfahren sowie ihre Realisierung in einem Softwarewerkzeug näher erläutert.

Bild 1.1 zeigt den Ablauf der Modellierung und Untersuchung eines Fertigungssystems mit der vorgeschlagenen Methode. Ausgangspunkt sind die für die Modellierung notwendigen Daten, die z.B. aus einer Layoutskizze des Fertigungssystems, einer Beschreibung der Eigenschaften der Betriebsmittel und den Arbeitsgängen bestehen können. Auf der linken Seite ist der funktionale und auf der rechten Seite der strukturelle Teil der Beschreibung dargestellt.

Für die einfache und komfortable Modellierung von Fertigungssystemen wurde eine spezielle Art farbiger Petri-Netze definiert. Damit ist der Modellierer in der Lage, Modelle der Struktur und der Arbeitspläne zu entwerfen. Dieser Aspekt ist bei der Modellierung mit Petri-Netzen bisher nicht ausreichend beachtet worden. Durch die Beschränkung der verwendbaren Typen (Farben) von Marken ist es nicht mehr notwendig, umfangreiche textuelle Beschriftungen und programmähnliche Konstrukte anzugeben. Die Verständlichkeit und Übersichtlichkeit der Modelle wird dadurch verbessert.

Mit diesen Petri-Netzen können die Modelle der Struktur und der Arbeitspläne direkt erstellt werden. Die alternative Modellierung eines Fertigungssystems mit Funktionssymbolen ermöglicht es jedoch auch Entwicklern ohne die Kenntnis von Petri-Netzen, die entwickelte Methode zu verwenden. Dabei wird ebenfalls zwischen Struktur des Fertigungssystems und den Arbeitsplänen getrennt. Diese symbolischen Modelle können automatisch in ein spezielles Petri-Netz umgewandelt werden.

Um oft auftretende Modellteile nicht immer wieder neu spezifizieren zu müssen, wird eine erweiterbare Modulbibliothek zur Verfügung gestellt. Einander ähnliche Teilmodelle können aus demselben Grundmodell abgeleitet werden. Dazu werden die in der Bibliothek enthaltenen Modelle mit den gewünschten Parametern instanziiert und in das Modell eingesetzt. Die Modulbibliothek wird außerdem für die Erzeugung von speziellen Petri-Netzen aus symbolischen Modellen verwendet.

Aus den Modellen der Struktur und der Arbeitspläne wird dann automatisch ein Gesamtmodell berechnet. Dies enthält neben dem Strukturmodell die Informationen aus den verschiedenen Arbeitsplänen, die in sogenannten Übergangstabellen vom Modellierer versteckt enthalten sind. Es entsteht ein das gesamte Verhalten modellierendes spezielles Petri-Netz, welches im Anschluß qualitativ und quantitativ untersucht werden kann. Dazu wurden existierende Methoden für die spezielle Modellart angepaßt. Mit Hilfe der interaktiven Simulation kann die korrekte Funktionsweise des Modells überprüft werden.

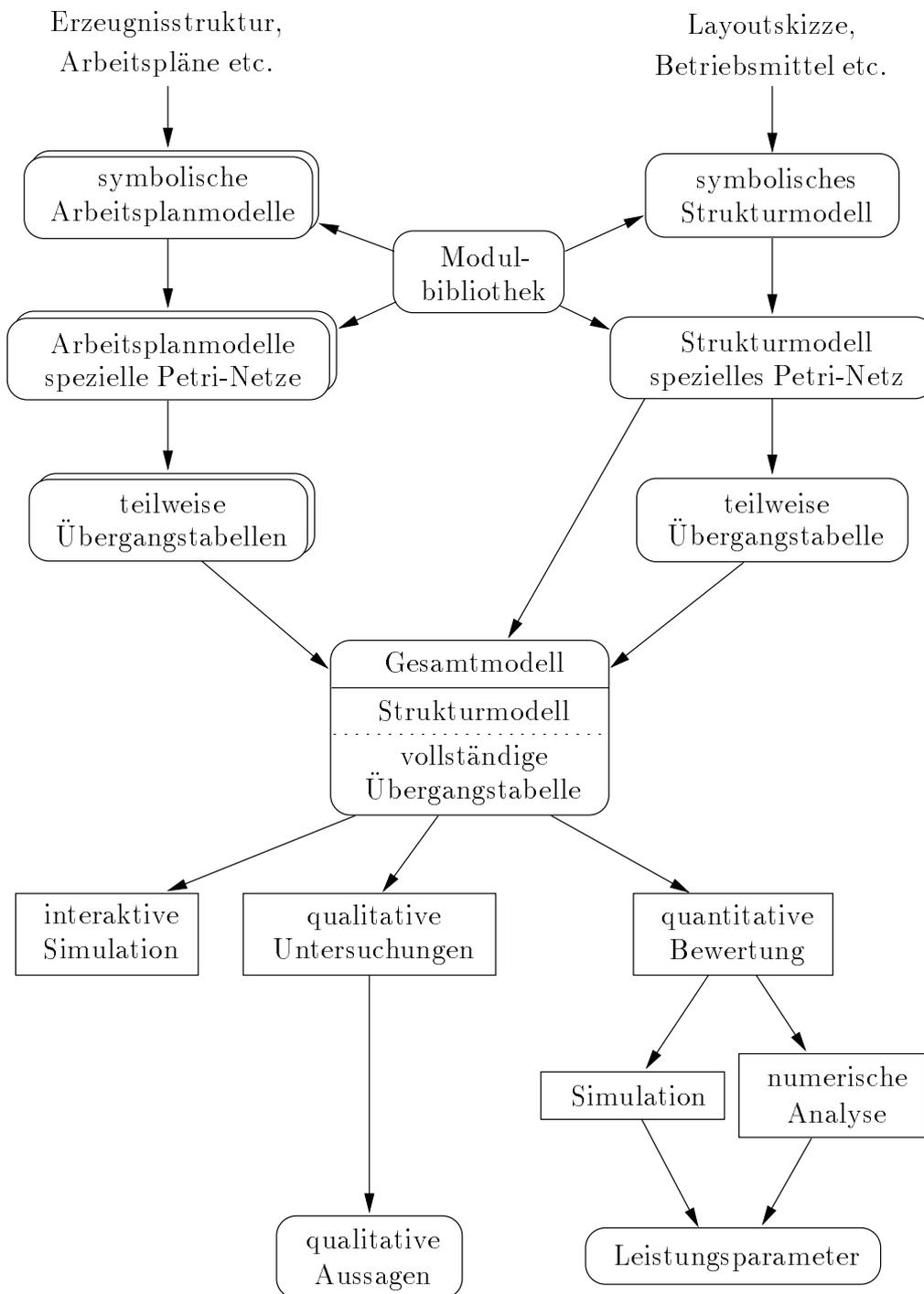


Abbildung 1.1: Überblick über die Modellierungs- und Bewertungsmethode

Die entwickelten Verfahren wurden in einem Softwarewerkzeug als Erweiterung von TimeNET [33] realisiert. Mit dessen Hilfe wurde die Leistungsfähigkeit der Methode an einem realen Anwendungsbeispiel nachgewiesen. Die eingesetzte Benutzungsoberfläche muß für verschiedene Modellarten verwendbar sein. Darum wurde eine parametrisierbare Benutzungsoberfläche für hierarchische Graphen entwickelt.

1.4 Aufbau der Arbeit

Im Anschluß an die Einführung wird im Kapitel 2 auf Probleme der Fabrikplanung und bekannte Lösungsansätze eingegangen. Dazu werden zunächst Anforderungen an Modellierungs- und Analysetechniken motiviert und genannt. Der folgende Abschnitt beschreibt ein fiktives Fertigungssystem, anhand dessen die entwickelte Modellierungsmethode und ihr Einsatz in Kapitel 3 geschildert wird. Außerdem werden Ausschnitte davon für eine Gegenüberstellung verschiedener aus der Literatur bekannter Modellierungsmethoden für Fertigungssysteme verwendet. Dies geschieht in den Abschnitten 2.3 bis 2.5 für klassische und farbige Petri-Netze sowie andere Beschreibungsmethoden. Den Abschluß des Kapitels 2 bildet eine Einordnung der vorliegenden Arbeit und Vergleich mit den vorher beschriebenen Techniken.

Im anschließenden Kapitel 3 wird die entwickelte Modellierungstechnik beschrieben. Grundlage dafür bildet die Definition der verwendeten Netzklasse, zu der danach einige Erläuterungen gegeben werden. Die wichtigsten Punkte der Modellierungsmethode werden in den folgenden Abschnitten 3.2 bis 3.4 anhand des fiktiven Beispiels dargestellt. Dazu gehört die getrennte Modellierung von Struktur und Arbeitsplänen sowie die automatische Erzeugung eines Gesamtmodells daraus. Außerdem wird das Vorgehen bei der Entwicklung von Bibliotheksmodulen und ihre spätere Verwendung in einem Modell beschrieben.

Das darauffolgende Kapitel 4 beschäftigt sich mit Techniken zur Ableitung von Aussagen über ein modelliertes Fertigungssystem. Zunächst werden häufig auftauchende Fragestellungen und ihre Umsetzung in Leistungsmaße des Modells besprochen. Danach wird erläutert, wie qualitative Eigenschaften eines speziellen Petri-Netzes bestimmt werden können. Den Anschluß bildet Abschnitt 4.3, der nach einer Darstellung des Modellverhaltens die Berechnung quantitativer Leistungsparameter durch Simulation oder numerische Analyse erläutert.

Kapitel 5 geht dann auf die alternativ mögliche Modellierung eines Fertigungssystems mit Funktionssymbolen ein. Anhand eines Beispiels wird die Vorgehensweise gezeigt. Thema des letzten Abschnitts des Kapitels ist die automatische Umwandlung eines symbolischen Modells in ein spezielles Petri-Netz. Die dabei entstehenden Modelle können dann direkt mit den vorher beschriebenen Methoden bewertet werden, so daß der Modellierer sich nicht mit dem darunterliegenden Petri-Netz befassen muß.

Anhand eines realen Anwendungsbeispiels wird in Kapitel 6 die Tauglichkeit der vorgeschlagenen Methoden überprüft. Das modellierte Fertigungssystem wird beschrieben und die zu beantwortenden Fragestellungen genannt. Auf die Darstellung des Modells folgen qualitative und quantitative Auswertungen. Die Ergebnisse werden erläutert und die Modellierungsmethode bewertet.

Die Realisierung der vorgestellten Methoden in einem Softwarewerkzeug beschreibt Kapitel 7. Für die verschiedenen auftretenden Modellarten wurde eine parametrisierbare Benutzungso-

berfläche entwickelt, mit der sich der erste Abschnitt beschäftigt. Ihre Verwendung während der Modellierung, Überprüfung und Bewertung eines Fertigungssystems sind ebenfalls Gegenstand dieses Kapitels. Außerdem wird auf die technische Umsetzung der Modellierungs- und Analysemethoden sowie Fragen der Implementierung eingegangen.

Das letzte Kapitel faßt die Ergebnisse zusammen und weist auf offene Probleme sowie mögliche weiterführende Arbeiten hin. Der Anhang enthält im Teil A eine Auflistung der verwendeten Symbole und ihrer Bedeutung. Teil B definiert Multimengen und ihre Eigenschaften, die bei der Definition der Netzklasse und der Beschreibung des Netzverhaltens benötigt werden. Danach sind die Formate von Modellbeschreibungen, Modellklassen der Benutzungsoberfläche, Übergangstabellen und in den Modellen auftretenden textuellen Elemente angegeben.

Kapitel 2

Modellierung von Fertigungssystemen

In diesem Kapitel werden verschiedene aus der Literatur bekannte Hilfsmittel zur Modellierung von Fertigungssystemen vorgestellt. Zunächst werden dabei auftretende Anforderungen und Problemstellungen genannt. Der zweite Abschnitt stellt das fiktive Beispiel eines Fertigungssystems vor, anhand dessen später die Modellierungsmethode der speziellen Petri-Netze erklärt wird. Im Anschluß werden verschiedene Beschreibungsmittel vorgestellt und in ihrer Eignung für die Modellierung von Fertigungssystemen bewertet. Dazu werden Ausschnitte des Fertigungssystem-Beispiels verwendet. Da sich diese Arbeit mit einer auf Petri-Netzen basierenden Modellierungsmethode beschäftigt, nimmt die Beschreibung der existierenden Methoden auf diesem Gebiet den größten Raum ein. Den Abschluß des Kapitels bildet eine Einordnung der vorliegenden Arbeit im Vergleich zu den vorher geschilderten bekannten Ansätzen.

2.1 Anforderungen und Problemstellungen

In der Einleitung wurde bereits begründet, daß die Entwicklung von Fertigungssystemen Beschreibungsmittel, Methoden und Werkzeuge zur Unterstützung des Planungsprozesses erfordert. Moderne Fertigungssysteme sollen sowohl effizient als auch anpassungsfähig sein [81]. Diese Ziele widersprechen sich teilweise, da eine hohe Flexibilität durch große Lagerbestände und geringe Auslastung der Betriebsmittel erreicht werden kann, was zu einer ineffizienten Fertigung mit hoher Kapitalbindung führt. Inzwischen hat sich die Erkenntnis durchgesetzt, daß optimale Lösungen dieses Problems nicht zu finden sind. Die untersuchten Optimierungsverfahren scheiterten an ihrer mathematischen Komplexität für realistische Systeme [44]. Statt dessen werden Strategien untersucht, die für bestimmte Problemstellungen akzeptable Lösungen mit vertretbarem Aufwand finden [70]. Dabei wurden unter anderem Methoden wie *push*, *pull*, *just in time*, Kanban und belastungsorientierte Fertigungssteuerung [80, 70, 44] vorgeschlagen.

Während der Planung eines Fertigungsprozesses muß geklärt werden, welche Produkte zu welchem Preis absetzbar sind. Daraus muß die Menge und Zusammensetzung des Produktmixes bestimmt werden, der mit den vorhandenen oder anzuschaffenden Betriebsmitteln wirtschaftlich sinnvoll produziert werden kann. Außerdem ist zu entscheiden, welcher Bedarf an Rohstoffen besteht. Für alle zu fertigenden Produkte sind Arbeitspläne aufzustellen,

die die notwendigen Arbeitsschritte und ihre Reihenfolge festlegen. Die Fertigungsablaufplanung legt darüber hinaus fest, zu welchem Zeitpunkt und in welcher Reihenfolge die verschiedenen Arbeitsgänge stattfinden sollen. Weiterhin müssen jeweils die verwendeten Betriebsmittel zugewiesen werden. Dabei kann ein fester Zeitplan (*schedule*) oder eine Menge von Steuerregeln (*scheduling strategy*) verwendet werden. Später muß die tatsächliche Produktionsdurchführung koordiniert und gesteuert werden. Dazu wird der Systemzustand überwacht und mit dem Sollzustand verglichen, um auf Abweichungen zu reagieren. Im Fehlerfall muß unter Umständen ein neuer Zeitplan erzeugt werden. Bei der Verwendung von Steuerregeln müssen diese anhand des Ist-Zustands des Fertigungsprozesses ausgewertet werden, um eventuell steuernd einzugreifen. Auf den darunterliegenden Sturzebenen werden die einzelnen Betriebsmittel lokal überwacht und beeinflußt [44, 70].

Um die Leistungsfähigkeit eines Fertigungssystems und seiner Steuerstrategien bewerten zu können, sind abhängig von der Fragestellung unterschiedlich genaue Modelle notwendig. Eine wichtige Aufgabe des Modellierers ist es, die für das Verhalten entscheidenden Einflüsse zu bestimmen und im Modell abzubilden. Für die Schnelligkeit der Berechnung von Leistungsgrößen ist das Weglassen unwichtiger Details ebenfalls von Bedeutung, da sonst die Berechnungsalgorithmen unnötig komplex werden. Die oben beschriebenen Fragestellungen bei der Produktionsplanung und -steuerung hängen eng zusammen und sind kaum voneinander losgelöst zu betrachten. Es ist daher sehr schwierig, die Auswirkungen einer gewählten Strategie auf den Fertigungsprozeß sicher abzuschätzen.

Andererseits werden die geplanten Fertigungssysteme komplexer und erfordern hohe Investitionssummen, während die zu erzielenden Gewinnspannen aufgrund des globalen Wettbewerbs kleiner werden. Darum ist die schnelle und sichere Vorherbestimmung der Leistungsgrößen eines geplanten Fertigungsprozesses von hoher Wichtigkeit. Da diese Untersuchungen nur an einem rechnerinternen Modell gemacht werden können, ist ein leicht zu bedienendes Werkzeug notwendig, welches die Schritte der Modellerstellung, Überprüfung und Leistungsbewertung integriert.

Die einem solchen Werkzeug zugrundeliegende Modellierungsmethode muß einfach und verständlich sein. Sie sollte demnach eine grafische Darstellungsform haben, die sich möglichst weit an dem modellierten System orientiert. Dabei sollte sich die Eingabe und Auswertung an der Terminologie der Fertigungsplanung orientieren. Die Modellierungsmethode darf nicht auf spezielle Fertigungsstrukturen eingeschränkt sein. Weitgehend unabhängige Aspekte sollten getrennt voneinander angegeben werden. Die getrennte Modellierung von Struktur und Arbeitsplänen sowie hierarchischer Modellaufbau sind demnach wichtig. Eine Bibliothek von Teilmodellen ist für die Wiederverwendbarkeit von oft auftretenden Modellteilen nützlich. Ihre Elemente können dann nach einer Belegung von Parametern instanziiert und verwendet werden. Dies verringert die Wahrscheinlichkeit fehlerhafter Modelle insbesondere in größeren Modellen. Die mit Hilfe der Modellierungsmethode gewonnenen Modelle müssen eindeutig und analysierbar sein.

Mit der Modellierungsmethode sollten Wechselwirkungen zwischen Werkstücken und Betriebsmitteln sowie Produkt- und Informationsfluß darstellbar sein. Verschiedene Entwurfstrategien (wie *top-down* oder *bottom-up*) sind anwendbar. Wiendahl nennt zusammengefaßt die folgenden Anforderungen [81]: Benutzereffizienz, Flexibilität der Anwendung, Benutzerfreundlichkeit, Programmierbarkeit, Eindeutigkeit, Interpretierbarkeit, Detaillierbarkeit, Manipulierbarkeit und Analysierbarkeit.

Ein nicht zu unterschätzendes Problem bei der Modellerstellung besteht in der Auswahl und Gewinnung der notwendigen Daten. Falls Veränderungen an einem bestehenden Fertigungssystem untersucht werden sollen, können diese Werte gemessen werden. Schwieriger ist dies für noch nicht existierende Systeme. Da die Modellierung und Leistungsbewertung letztendlich aus dem Wissen um lokale Eigenschaften (Bearbeitungszeiten, Puffergrößen) globale Fragestellungen (Durchsatz) beantwortet, müssen spezifische Daten für die modellierten Betriebsmittel beschafft werden. Einige dieser Einflußgrößen sind direkt beeinflussbar, wie z.B. die Größe eines Puffers, oder aus dem geplanten System bestimmbar wie die Länge eines Förderbandes. Daten wie die Bearbeitungszeiten von Fertigungsaufträgen sowie Ausfall- und Reparaturzeiten müssen anhand von Erfahrungen oder Datenblättern der Hersteller bestimmt werden. Diese Daten sind aber zum Teil auch für die Produktionsplanung erforderlich und können eventuell aus bereits aufgestellten Arbeitsplänen o.ä. entnommen werden.

Ebenso wie bei der Modellierung eines Fertigungssystems von unwichtigen strukturellen Dingen abstrahiert wird, muß auch bei der Angabe der Schaltzeitverteilungen in einem zeitbehafteten Modell sinnvoll vereinfacht werden. Für eine Fallstudie wird dies im Kapitel 6 durchgeführt. Vergleichende Modellauswertungen können Hinweise darauf geben, wie stark die Ergebnisse von der Verteilung der Schaltzeit abhängen. Die Wahl einer bestimmten Schaltzeitverteilung kann großen Einfluß auf die Komplexität der Bestimmung der Leistungsparameter haben.

Für die praktische Anwendbarkeit einer Modellierungsmethode muß diese als Softwarewerkzeug realisiert sein, welches die entsprechenden Techniken unterstützt. Dieses sollte einfach und intuitiv zu bedienen sein und eine leichte grafische Manipulierbarkeit des Modells erlauben. Das ist auch für einen effizienten Modellervorgang wichtig, der typischerweise aus dem mehrfachen Durchlaufen von Modellieren bzw. Verändern, Überprüfen des Modells sowie der Leistungsbewertung besteht. Soweit möglich, sollten Eingaben auf syntaktische Plausibilität überprüft bzw. offensichtlich falsche Daten zurückgewiesen werden. Das Werkzeug sollte Schnittstellen zum Datenaustausch mit anderen Programmen besitzen und den Modellierer besonders bei Routineaufgaben entlasten. Für systematische Vergleiche verschiedener Modelle kann die automatische Auswertung und grafische Darstellung der Ergebnisse nützlich sein.

Nach oder schon während der Eingabe eines Modells muß der Modellierer in der Lage sein, das Modell zu überprüfen. Dazu kann getestet werden, ob das Modell den Anforderungen der Modellierungsmethode entspricht und plausibel ist. Wichtig für das „Vertrauen“ des Modellierers in Ergebnisse und das bessere Verständnis des Fertigungssystems ist die Möglichkeit der interaktiven Simulation. Dabei werden innerhalb des Softwarewerkzeugs neben dem Zustand des Modells die möglichen Aktivitäten angezeigt. Der Modellierer wählt aus, welche davon ausgeführt werden soll, und kann so das Verhalten des Modells ausprobieren und mögliche Fehler finden. Mit Hilfe der qualitativen Untersuchung des Modells sind Eigenschaften wie z.B. die Verklemmungsfreiheit überprüfbar, ohne deren Erfüllung eine anschließende Leistungsbewertung möglicherweise sinnlos ist. Neben der Berechnung von Leistungsparametern ist für große Modelle auch die Bestimmung von Stücklisten, Teilverwendungsnachweisen, Arbeitsplänen, Ein- und Ausgängen des Systems, unbenutzter Betriebsmittel u.a. aus dem fertigen Modell zur Überprüfung sinnvoll.

Neben der Spezifikation des eigentlichen Modells müssen für eine Leistungsbewertung auch die interessierenden Leistungsparameter angegeben werden können. Diese sollten möglichst

in das Modell integriert sein. Häufig auftretende Fragestellungen müssen einfach angebar sein, aber auch komplexe Ausdrücke sind notwendig. Wichtige Zielgrößen sind z.B. der Durchsatz bestimmter Produkte, der Maschinennutzungsgrad, Pufferfüllstände, Durchlaufzeiten, Warte-, Bearbeitungs-, und Transportzeiten von Werkstücken, Fertigungskosten sowie Ausfallzeiten von Betriebsmitteln. Auf diesem Weg können dann auch indirekt Engpässe des Systems gefunden werden. Durch den Vergleich verschiedener Varianten des Modells und ihre Bewertung wird eine Optimierung erreicht.

Zur Berechnung der Leistungsgrößen müssen effiziente numerische Methoden (Simulation oder Analyse) existieren. Die verwendeten Algorithmen sollten möglichst keine tiefgreifenden Kenntnisse des Modellierers voraussetzen und numerisch stabil sein. Die gewünschte Genauigkeit sollte auswählbar sein und durch die Anwendung der Algorithmen nachweisbar erreicht werden.

2.2 Ein Beispiel-Fertigungssystem

Dieser Abschnitt beschreibt ein fiktives Fertigungssystem, anhand dessen innerhalb dieses Kapitels unterschiedliche Beschreibungsmethoden für Fertigungssysteme einander gegenübergestellt werden. Das Beispiel wird außerdem in späteren Kapiteln verwendet, wenn Einzelheiten der in dieser Arbeit vorgestellten Methoden erläutert werden. Ein größeres, reales Anwendungsbeispiel und seine Modellierung und Untersuchung wird im Kapitel 6 beschrieben.

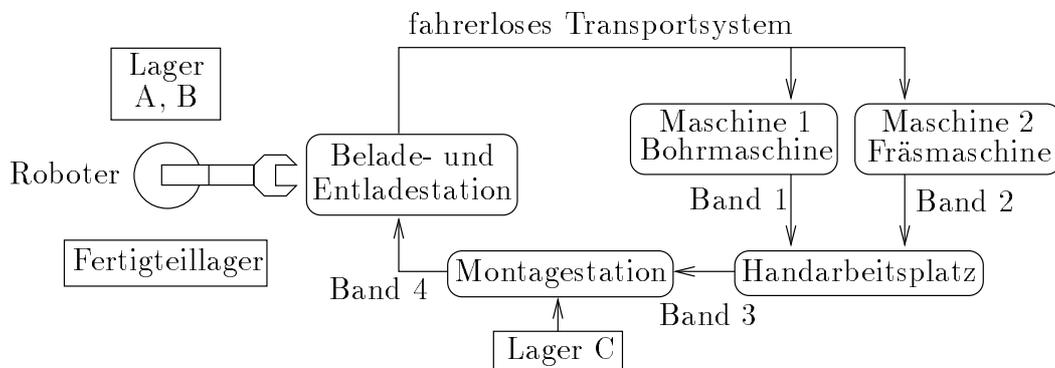


Abbildung 2.1: Aufbau des Beispiel-Fertigungssystems

Bild 2.1 zeigt eine Layoutskizze des Beispiel-Fertigungssystems. Es handelt sich um eine Fertigungszelle mit zwei automatischen Maschinen bzw. Fertigungszellen (Bohr- und Fräsmaschine), einer Montagestation und einem Handarbeitsplatz. Die Werkstücke werden auf je einem Werkstückträger durch das System transportiert. Das Be- und Entladen der Werkstückträger erfolgt durch den Roboter an der Ladestation. Die möglichen Transportwege der Werkstückträger zwischen den einzelnen Stationen sind durch Pfeile gekennzeichnet. Lager C befindet sich direkt an der Montagestation, so daß hier kein Transport notwendig ist. Den Transport der Werkstückträger von der Ladestation zu den Maschinen 1 und 2 wird durch ein fahreres Transportsystem (FTS) mit einem Fahrzeug übernommen. Ist das Fahrzeug leer, bewegt es sich automatisch zur Ladestation, um einen Werkstückträger

<i>Ort</i>	<i>Kapazität</i>	<i>Bemerkungen</i>
Lager A, B	1	
Lager C	4	
Fertigteillager	2	
Be- und Entladestation	1	
Maschine 1 (Bohrmaschine)	1	kann Ausfallen
Maschine 2 (Fräsmaschine)	1	kann Ausfallen
Handarbeitsplatz	1	Prüfen, Entgraten
Montagestation	1	
FTS (Ladestation→M1 und M2)	1	ein Fahrzeug
Förderband 1 (M1→Handarbeit)	2	
Förderband 2 (M2→Handarbeit)	2	
Förderband 3 (Handarbeit→Montage)	1	
Förderband 4 (Montage→Ladestation)	1	

Abbildung 2.2: Betriebsmittelliste

aufzunehmen. Ist dies geschehen, fährt es je nach der zu erfüllenden Transportaufgabe eine der Maschinen 1 oder 2 an, um dort wieder entladen zu werden. Für die anderen Verbindungen sind insgesamt vier voneinander unabhängige Förderbänder verantwortlich. Die Förderbänder bewegen auf ihnen befindliche Werkstückträger kontinuierlich bis zum Bandende bzw. solange, bis sie durch vor ihnen liegende Werkstückträger blockiert werden.

Die unbearbeiteten Teile der Typen A, B und C werden zu den entsprechenden Lagern (Lager A, B sowie Lager C) angeliefert. Fertiggestellte Werkstücke nimmt der Roboter vom Werkstückträger und legt sie in das Fertigteillager, von wo sie abtransportiert werden. In der Betriebsmittelliste (Bild 2.2) sind je nach Aufgabenstellung verschiedene Angaben über die vorhandenen Maschinen, Transportmittel usw. enthalten. Dazu zählt z.B. die Kapazität, die Geschwindigkeit bzw. die Bearbeitungsdauer sowie das Stör- und Reparaturverhalten. In den Stationen kann jeweils nur ein Werkstück auf einmal bearbeitet werden.

Die in Bild 2.3 dargestellten Erzeugnisstrukturbäume [44] stellen die qualitative und quantitative Zusammensetzung der Produkte A und B dar, die in unserem Beispiel gefertigt werden sollen. Produkt A wird demnach aus einem angelieferten Grundteil A und zwei Teilen vom Typ C gefertigt, während für B neben dem Grundteil ein zusätzliches Teil C notwendig ist. Durch das Fertigungssystem werden zwei Typen von Werkstücken gefertigt, die als A und B bezeichnet werden. Bild 2.4 zeigt von oben nach unten die notwendigen Arbeitsschritte zur

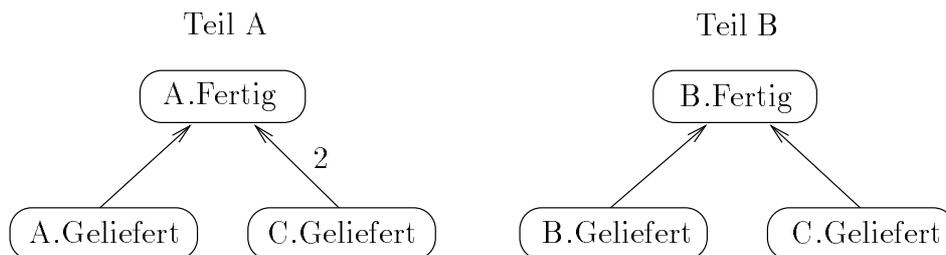


Abbildung 2.3: Erzeugnisstrukturbäume Werkstücke A und B

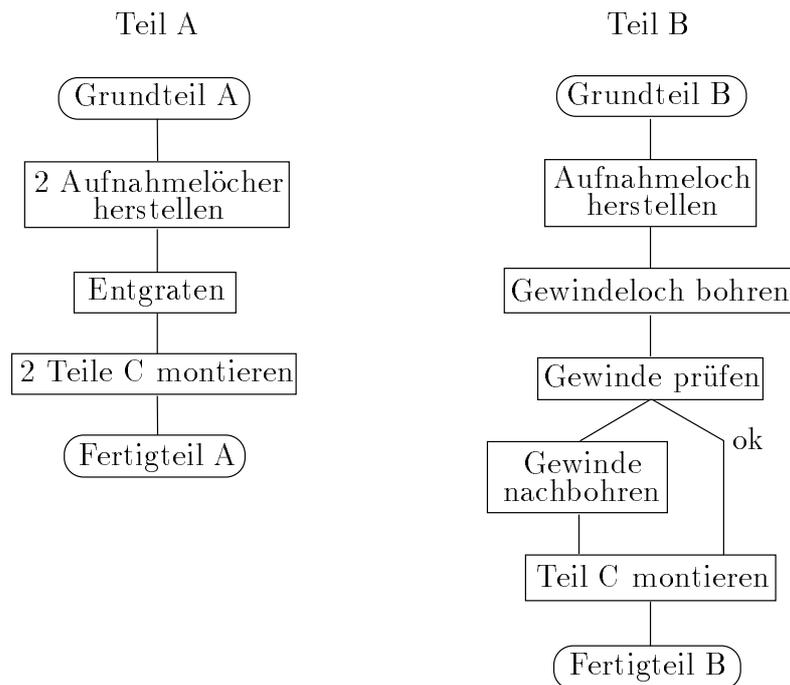


Abbildung 2.4: Arbeitsschritte der Werkstücke A und B

Fertigung der Teile aus den angelieferten unbearbeiteten Werkstücken.

Im Grundteil A müssen zunächst zwei Aufnahmeelöcher hergestellt werden. Nach dem Entgraten wird in jedem Aufnahmeeloch ein Teil C befestigt. Werkstück A ist damit fertiggestellt. Im Grundteil B muß als erstes ein Aufnahmeeloch hergestellt und ein weiteres Loch mit Gewinde gebohrt werden. Das Gewinde wird geprüft und muß nachgebohrt werden, falls es noch nicht den gestellten Anforderungen entspricht. Danach wird ein Teil vom Typ C montiert, und das Werkstück ist fertig.

Als nächstes muß entschieden werden, welche der im Fertigungssystem verfügbaren Betriebsmittel die eben beschriebenen Arbeitsschritte ausführen sollen. Dabei entstehen die in Bild 2.5 von oben nach unten dargestellten Arbeitspläne. Hier sind nicht nur die einzelnen Arbeitsschritte und die dafür vorgesehenen Maschinen verzeichnet, sondern auch die notwendigen Transporte zwischen den Stationen. In jedem Kästchen steht die auszuführende Aufgabe über dem verantwortlichen Betriebsmittel; Transporte sind durch das Symbol „→“ gekennzeichnet. Arbeitspläne werden oft auch in Tabellenform dargestellt und enthalten neben der Strukturierung einer Herstellungsaufgabe weitere Informationen wie Bearbeitungszeiten [26], die hier aus Platzgründen weggelassen wurden. Aus ihnen lassen sich die notwendigen Schritte, ihre Reihenfolge und eventuelle Alternativen ablesen. Für jeden Arbeitsschritt werden das beteiligte Betriebsmittel, die verwendeten Ausgangsmaterialien und entstehenden Werkstücke spezifiziert. Alternative Abfolgen von Fertigungsschritten können dadurch entstehen, daß derselbe Arbeitsgang mit unterschiedlichen Maschinen oder Transportmitteln durchgeführt werden kann. Im Beispiel kann außerdem abhängig vom Ergebnis einer Werkstückprüfung eine Route mit zusätzlichen Arbeitsschritten notwendig sein. Weitere Grunddaten und Datenstrukturen, die bei der Produktionsplanung und -steuerung auftreten, sind in [44] beschrieben.

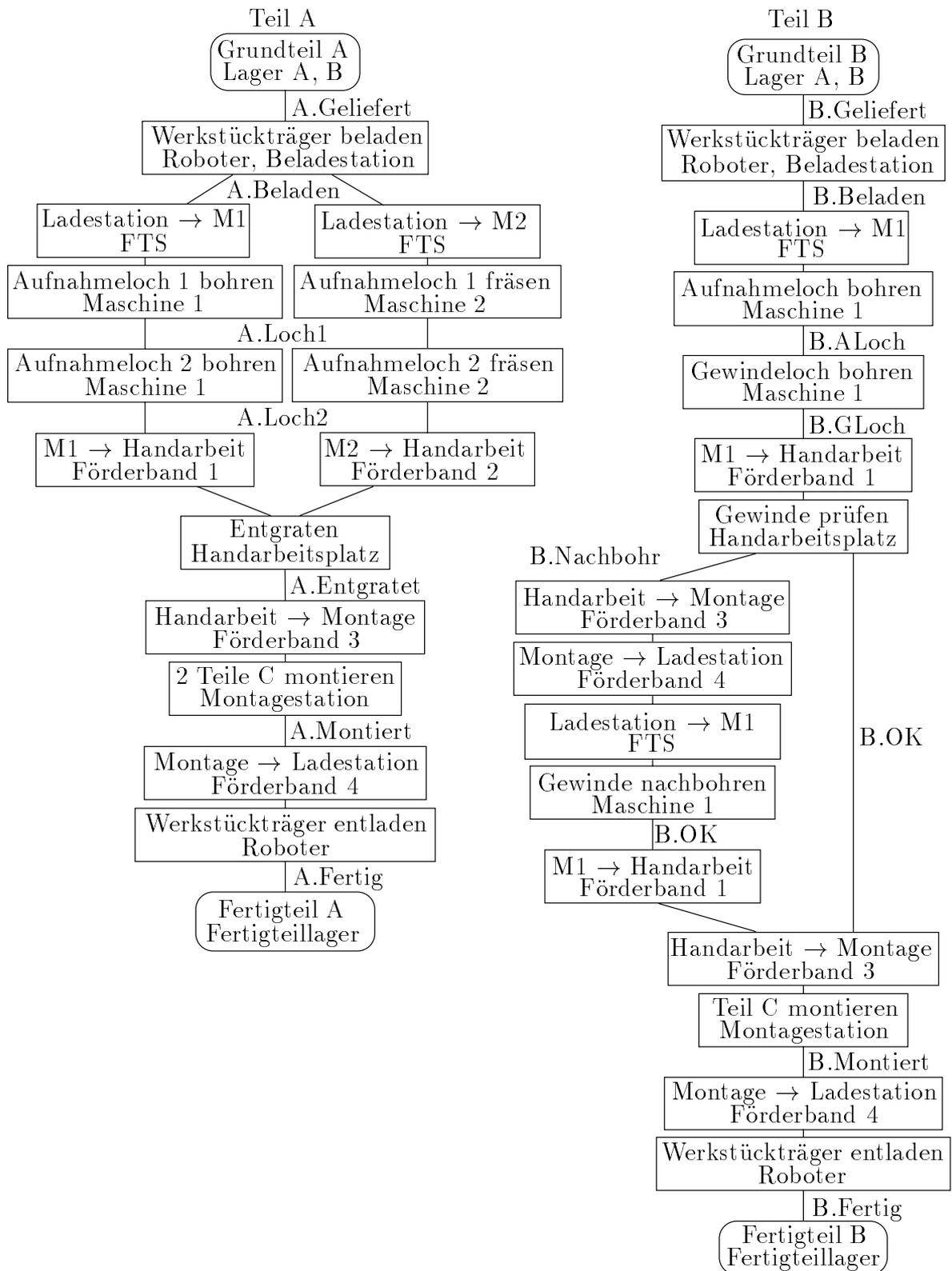


Abbildung 2.5: Arbeitspläne für das Beispiel-Fertigungssystem

Die Werkstücke durchlaufen während ihrer Bearbeitung verschiedene Zustände, denen zu ihrer späteren Unterscheidung Namen zugewiesen werden. Dabei werden der Name des Werkstücks und der Bearbeitungsstufe durch einen Punkt getrennt angegeben. Zwischen einigen Kästchen stehen die Bezeichnungen der Werkstücke in den verschiedenen Bearbeitungsstufen. Eine neue Bezeichnung eines Bearbeitungszustands steht dabei immer direkt unter dem Arbeitsgang, der das Werkstück in diesen Zustand bringt. Fehlt diese Bezeichnung, ändert sich der Zustand durch den vorangegangenen Arbeitsgang nicht und es gilt weiterhin der vorherige Zustand. Dies ist speziell bei den reinen Transportaufgaben immer der Fall. Da abhängig vom Ort eines Werkstücks eindeutig festgelegt ist, ob es sich auf einem Werkstückträger befindet oder nicht, wird nur der Werkstückname angegeben.

Alle Montagearbeiten werden durch die Montagestation übernommen, während am Handarbeitsplatz entgratet und geprüft wird. Der Roboter übernimmt das Be- und Entladen der Werkstückträger in der Ladestation. Die Herstellung der zwei Aufnahmelöcher bei Teil A kann sowohl durch die Bohrmaschine als auch durch die Fräsmaschine erfolgen. Darum trennt sich der Arbeitsplan hier in zwei Varianten auf. Welche gewählt wird, soll von der aktuellen Auslastung der beiden Maschinen abhängen.

Bei der Fertigung von Teil B muß ein Aufnahme- und ein Gewindeloch gefertigt werden. Das Gewindeloch kann nur durch die Bohrmaschine hergestellt werden. Um unnötige Transporte zu vermeiden, wird danach das Aufnahmeloch ebenfalls mit der Bohrmaschine gefertigt. Abhängig vom Prüfergebnis am Handarbeitsplatz muß Teil B nachgebohrt werden. Dafür wird der entsprechende Werkstückträger wieder zur Bohrmaschine gebracht. Leere Werkstückträger werden ebenfalls durch das System transportiert, um nicht den Weg für die Werkstücke zu versperren. Der entsprechende Transportplan ist trivial und wird darum hier nicht dargestellt; nur beim Transport durch das FTS ist eine Auswahl zwischen den Zielen Maschine 1 und Maschine 2 möglich.

2.3 Klassische Stochastische Petri-Netze

Die Modellierung von Fertigungssystemen ist eines der ältesten Anwendungsgebiete von Petri-Netzen. Dementsprechend gibt es viele Arbeiten auf diesem Gebiet, in denen (farblose) Petri-Netze bzw. Stellen-Transitions-Netze [59] und ihre stochastischen Erweiterungen wie SPNs [2], GSPNs [4, 16] und DSPNs [5] zu diesem Zweck verwendet werden. Ein Überblick kann [69, 70, 77] entnommen werden. Verschiedene Anwendungsbereiche auf dem Gebiet der Automatisierungstechnik beschreiben die Arbeiten in [64].

Sie eignen sich gut für die Abbildung von Fertigungssystemen, da sie auf grafisch einfache Weise auch komplexe Abläufe darstellen können. Die Gewinnung großer Modelle ist durch schrittweise Verfeinerung möglich [83]. Eine große Anzahl von Verfahren und Werkzeugen zur Untersuchung von Petri-Netz-Modellen wurde in der Vergangenheit entwickelt.

Ein Vergleich von GSPNs und Warteschlangenmodellen bei der Leistungsbewertung von Fertigungssystemen wurde in [9] vorgenommen. Für Fertigungssysteme, in denen Prioritäten und Synchronisation eine Rolle spielen, sind GSPNs überlegen. Der zu untersuchende Erreichbarkeitsgraph wird jedoch schnell sehr groß.

Zurawski und Dillon untersuchten Methoden zur systematischen Konstruktion typischer Teile eines flexiblen Fertigungssystems [90]. Die vorgestellte Technik hilft bei der notwendigen Aufspaltung bzw. Entfaltung von Stellen, die verschiedene Produkte enthalten können. Das bei der Modellierung von Fertigungssystemen mit farblosen Petri-Netzen auftretende Problem, in denen verschiedene Werkstücke unterschieden werden müssen, wird hier besonders deutlich. Die so gewonnenen Modelle werden zur Verifikation des Modellverhaltens verwendet.

In [6] werden Bausteine für den modularen Aufbau von GSPN-Modellen für Fertigungssysteme beschrieben. Dadurch wird der Modellierungsvorgang einfacher und weniger fehleranfällig. Spezielle Symbole werden für oft auftretende Konstrukte verwendet und später durch kleine Petri-Netz-Modellteile ersetzt. Nach einer strukturellen Analyse, die die Lebendigkeit und Beschränktheit der Modelle prüft, wird eine Leistungsbewertung vorgenommen. Durch die Verbindung von Programmcode mit den Stellen des Modells, der bei Ankunft einer Marke ausgeführt wird, ist eine Steuerung des realen Fertigungssystems möglich.

Mit der *bottom-up*-Synthese von Petri-Netz-Modellen beschäftigen sich [84, 85]. Dabei steht die Erhaltung der Lebendigkeit und Beschränktheit der Modelle während der schrittweisen Erweiterung im Vordergrund. Eine besondere Rolle spielen dabei gemeinsam genutzte Betriebsmittel, durch die Verklemmungen (*deadlocks*) unbeabsichtigt im Modell entstehen können. Bei der Beachtung entsprechender Regeln während der Kombination von Teilmodellen zu größeren Blöcken können die Invarianten des Modells direkt aus denen der Teilmodelle abgeleitet werden.

Die Feinplanung (*scheduling*) von flexiblen Fertigungssystemen mit Hilfe stochastischer Petri-Netze und heuristischer Regeln ist Gegenstand einiger Arbeiten wie z.B. [36, 46]. Die Leistungsfähigkeit verschiedener Regelsysteme wird dabei miteinander verglichen. In der zweiten Arbeit wird dagegen der Erreichbarkeitsgraph teilweise durchsucht, um annähernd optimale Schaltfolgen von Transitionen und damit eine gesuchte Abfolge von Aktivitäten zu finden.

Verschiedene Autoren beschäftigten sich mit der Verwendung nichtexponentieller Schaltzeiten in Modellen von Fertigungssystemen. In [79] werden Transitionen, für die Mittelwert und Varianz der Schaltzeitverteilung bekannt sind, durch ein entsprechend parametrisiertes Teilmodell nachgebildet, das exponentielle Transitionen enthält. Ein DSPN-Modell eines Fertigungssystems wird in [49] beschrieben und in seiner Leistungsfähigkeit untersucht.

Beck und Krogh schlugen verschiedene Typen von Stellen in einem farblosen Modell vor, um zwischen sogenannten *resource states* und *operation states* unterscheiden zu können [10]. Ein Fertigungsprozeß wird in Operationen mit festgelegter Reihenfolge zerlegt. Für jede Operation sind die beteiligten Betriebsmittel und ihre Zustände zu spezifizieren, die durch die Operation verändert werden können. Sogenannte *modified Petri nets* entstehen durch die Kombination von Teilmodellen der Zustände von Betriebsmitteln. So werden systematisch Modelle von Fertigungssystemen konstruiert.

In Bild 2.6 ist ein kleiner Teil des in Abschnitt 2.2 vorgestellten Beispiel-Fertigungssystems mit einem GSPN modelliert. Dabei wurde aus Platzgründen nur der Roboter mit seinen Verbindungen dargestellt und dabei lediglich der Arbeitsplan für das Produkt A berücksichtigt. Zusammengehörige Modellkomponenten sind mit gestrichelten Rechtecken zusammengefaßt. Zur Illustration der Verbindung mit benachbarten Modellteilen ist außerdem noch die Anlieferung der Vorprodukte in das Eingangslager dargestellt.

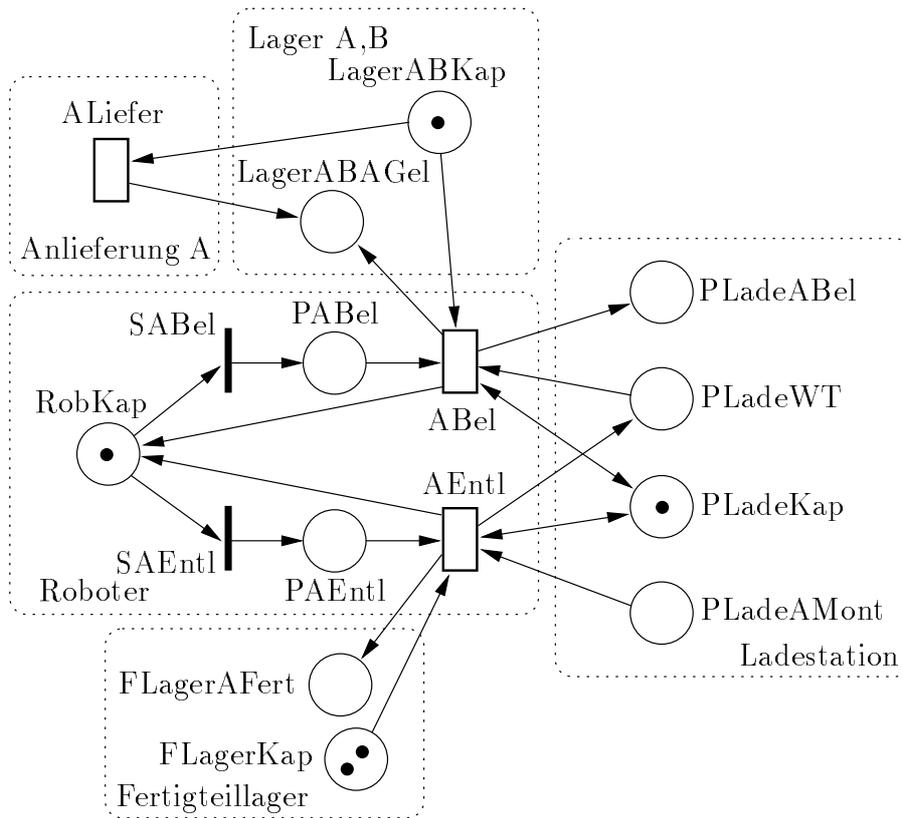


Abbildung 2.6: Farbloses Teilmodell des Roboters

Lager und Puffer werden durch Stellen abgebildet. In jedem Lager muß zwischen den verschiedenen Produktarten unterschieden werden. Dazu wird jedes Lager mit so vielen Stellen modelliert, wie unterschiedliche Produkte in ihm auftreten können [90]. Im Beispiel sind die Stellen `PLadeABel` für Werkstücke des Typs `A.Beladen`, `PLadeWT` für `WT.leer` und `PLadeAMont` für `A.Montiert` in der Be- und Entladestation notwendig. Aufgrund dieser Entfaltung der Stellen ist es nicht mehr möglich, Kapazitäten direkt an einer Stelle anzugeben. Darum müssen Kontrollstellen eingeführt werden, deren initiale Markierung der Kapazität des jeweiligen Lagers entspricht. Für komplexere Kapazitäten, die z.B. von der Produktart abhängen, wäre das nicht so einfach möglich. Im Beispiel ist `PLadeKap` für die Beschränkung der Kapazität der Be- und Entladestation eingeführt worden.

Ein Betriebsmittel wie der Roboter ist für mehrere Arbeitsaufgaben verantwortlich, im Beispiel für das Beladen der Werkstückträger und spätere Entladen der fertigen Teile. Der Roboter kann nur eine Aufgabe auf einmal ausführen, und jede spezielle Aufgabe benötigt möglicherweise eine andere Zeit. Darum wird der Roboter wie im Bild gezeigt durch so viele Pfade Transition-Stelle-Transition modelliert, wie er Aufgaben hat. Dies wäre in diesem Beispiel eigentlich nicht nötig, da durch die Kapazität von Eins der Be- und Entladestation die Aufgabe des Roboters immer eindeutig festgelegt ist, aber es soll die prinzipielle Vorgehensweise beschrieben werden. Die Stelle `RobKap` und die nachfolgenden zeitlosen Transitionen `SABel` und `SAEntl` sichern, daß höchstens eine der eigentlichen Aktivitäten auf einmal ablaufen kann. Problematisch ist dabei die Auswahl der zu aktivierenden zeitbehafteten Transition durch das Schalten einer der zeitlosen Transitionen, denn diese sind ja bei verfügbarem Ro-

boter immer schaltfähig. Die Eingangskanten der zeitbehafteten Transitionen können aber nicht einfach vorverlegt werden, denn dadurch kann die Belegung der angrenzenden Stellen unsinnig werden. Für ein korrektes Verhalten müßten die zeitlosen Transitionen also noch zusätzliche Schaltbedingungen erhalten.

Die praktische Anwendbarkeit der farblosen Petri-Netze ist offensichtlich auf kleine Systeme eingeschränkt, in denen möglichst keine verschiedenen Produkte an derselben Stelle auftreten sollten. Sie werden sehr schnell unübersichtlich, da unterschiedliche Produkte nicht durch verschiedene Marken dargestellt werden können. Die Entfaltung von Stellen und Transitionen führt zu sehr großen Modellen, deren Struktur stark von den Arbeitsplänen und den auftretenden Werkstückarten abhängen und nicht mehr den wirklichen Aufbau verdeutlichen. Sie macht außerdem die Einführung von Kontrollstellen für Kapazitäten notwendig. Die typischerweise in Fertigungssystemen auftretenden exklusiven Betriebsmittel erfordern zusätzlichen Modellierungsaufwand. Für größere Modelle ist das Fehlen einer hierarchischen Darstellung nachteilig. Die Spezifikation von Leistungsparametern ist in einem farblosen Modell selten direkt möglich, da das Modell nicht der realen Struktur entspricht. Der Durchsatz einer Maschine muß beispielsweise über die sie modellierenden verschiedenen Transitionen berechnet werden. Vorteilhaft ist dagegen die rein grafische Darstellung sowie die einfachen und leicht verständlichen Modellelemente. Ein größeres Beispiel eines farblosen Modells einer Fertigungszelle und die dabei auftretenden Probleme sind in [86] dargestellt.

2.4 Farbige Petri-Netze

Ebenso wie für klassische Petri-Netze ist die Modellierung von Fertigungssystemen ein häufiges Anwendungsgebiet von *high-level* Petri-Netzen. Durch die Verfügbarkeit unterscheidbarer Marken können kompaktere Modelle erstellt werden. Die vorliegende Arbeit orientiert sich an den farbigen Petri-Netzen, deren Definition [40] entnommen werden kann. Der Übergang zu farbigen Petri-Netzen bedeutet keine Erweiterung der Modellierungsmächtigkeit gegenüber farblosen Modellen. Jedes farbige Modell kann, ebenso wie die in dieser Arbeit verwendeten speziellen Netze, in ein farbloses Modell mit gleichem Verhalten umgewandelt werden [40, 13]. Die Beschreibung selbst findet aber auf einem höheren Niveau statt. Der Modellierer kann Details spezifizieren, ohne den Überblick zu verlieren. Außerdem eröffnet sie die Möglichkeit, Teilmodelle wiederzuverwenden.

Durch die Möglichkeiten, Datenstrukturen mit Marken zu verknüpfen, bieten sich farbige Netze nach der Modellierung auch zur Generierung der Steuerung von Fertigungssystemen und deren Validierung an. Über die Untersuchung der Invarianten eines farbigen Petri-Netz-Modells werden beispielsweise in [7, 76] Aussagen über die Lebendigkeit eines Fertigungssystems abgeleitet.

Ein farbiges Petri-Netz-Modell der Steuerung einer flexiblen Fertigungszelle wird in [41] entwickelt. Die Autoren zeigen, wie die Lebendigkeit des Modells anhand von Invarianten überprüft werden kann. Darüber hinaus kann das Modell ausgeführt werden und über spezielle Schnittstellen Nachrichten und Zustandsinformationen mit dem realen System austauschen. Mit Hilfe des Modells ist damit eine Steuerung und Anzeige des aktuellen Zustands der Fertigungszelle möglich. Eine weitere Methode zur Modellierung und Validierung einer Steuerung für Fertigungssysteme mit farbigen Petri-Netzen wird in [27] beschrieben.

Gentina und Corbeel schlugen *structured adaptive coloured Petri nets* für die Modellierung von Fertigungssystemen vor [21, 29]. Diese enthalten Elemente strukturierter Programmiersprachen sowie farbiger Petri Netze, die durch hemmende und markierungsabhängige Kanten etc. erweitert sind. Aus einer formalen Beschreibung der durch das System zu erfüllenden Aufgaben wird mit Methoden der künstlichen Intelligenz ein erstes farbiges Petri-Netz generiert. Nach einer anschließenden Spezifikation des Transportsystems entsteht ein Petri-Netz, welches zur Steuerung des Fertigungssystems eingesetzt wird. Die Beschreibung der Arbeitspläne ist im Modell nicht mehr sichtbar, da sie sich in einem gemeinsamen Modell mit der übrigen Systembeschreibung befinden. Das generierte Modell ist aufgrund der Art der automatischen Erzeugung für den Modellierer schwer verständlich.

In [53] wird gezeigt, daß farbige Petri-Netze eine geeignete Modellierungstechnik für komplexe nebenläufige Systeme darstellen. Dazu werden unabhängige Subsysteme identifiziert und getrennt modelliert. Die entstehenden Teilmodelle werden durch eine Fusion korrespondierender Transitionen miteinander gekoppelt, so daß ein Gesamtmodell entsteht. Nach eventuellen Vereinfachungen des Modells kann seine Struktur untersucht werden. Ein interpretiertes Modell entsteht durch die Beschriftung der Transitionen mit Kommandos einer Programmiersprache, um z.B. die Steuerung eines Fertigungssystems in das Modell mit einzubeziehen.

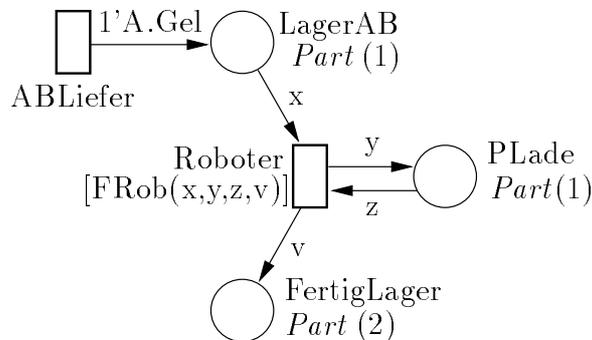
Martínez, Muro und Silva zeigen in [52], wie die Koordinationsebene eines flexiblen Fertigungssystems mit einem farbigen Petri-Netz abgebildet werden kann. Das Modell wird dann in die umgebenden Ebenen der Steuerung (lokale Steuerung und Scheduler) eingebettet, wobei eine auf den Markenfarben basierende Methode zur Interaktion verwendet wird. Die Analyse des Modells zeigt Verklemmungen und Entscheidungsprobleme auf. Außerdem wird eine Leistungsbewertung durchgeführt, um verschiedene Varianten des Fertigungssystems miteinander zu vergleichen. Für die zu fällenden Entscheidungen wird ein Scheduler vorgeschlagen, der Methoden der künstlichen Intelligenz verwendet.

Villaroel, Martínez und Silva entwickelten GRAMAN [75], ein Werkzeug zur Modellierung von Fertigungssystemen. In GRAMAN ist wie in der vorliegenden Arbeit die getrennte Modellierung von Arbeitsplänen und Struktur des Fertigungssystems möglich. Für die beiden Modellteile (*plant description* und *work plans*) werden unterschiedliche Beschreibungsmittel verwendet. Während für die Arbeitspläne farbige Petri-Netze gewählt wurden, erfolgt die Modellierung der Struktur des Fertigungssystems mit vordefinierten Bausteinen. Für jeden Baustein wird dann ein vordefiniertes Teilmodell mit den jeweiligen strukturellen Eigenschaften parametrisiert und eingesetzt, so daß ein internes Modell entsteht. Es gibt jedoch viele Fälle, in denen dies nicht ausreicht, da mehr Parameter für die Instanziierung eines Submodells notwendig sind. Die Eigenschaften einer Maschine hängen z.B. nicht nur von ihrer Struktur (den generellen Fähigkeiten) ab, sondern oft auch von einem speziellen Fertigungsauftrag. Beispielsweise kann die Bearbeitungsdauer auf einer Maschine variieren, je nachdem welches Produkt bearbeitet wird. Dies läßt sich mit GRAMAN nicht modellieren.

In GRAMAN werden die Verbindungen zwischen den Bausteinen bei der Erzeugung des internen Modells durch die Verschmelzung von Transitionen hergestellt, die synchrone bzw. gleichbedeutende Aktivitäten darstellen. Demnach müssen die Anzahl und die Art der möglichen Verbindungen einer Maschine mit ihrer Umgebung bereits bekannt sein, wenn das Teilmodell spezifiziert und in die Bausteinbibliothek übertragen wird. Dies widerspricht der Modularität, derzufolge auf der Ebene des Maschinenmodells nur Elemente der Maschine selbst angegeben werden sollten. Die sich ergebenden Teilmodelle spiegeln außerdem nicht mehr die

Struktur des betrachteten Systemteils wider, denn die Art und Weise des Entladens eines Maschinenpuffers sollte beispielsweise dort angegeben werden, wo der dafür verantwortliche Roboter modelliert wird. In GRAMAN können äquivalente Modellteile zusammengefaßt werden, was zwar zu einer kompakteren, aber nicht unbedingt verständlicheren Darstellung führt. Die Angabe von Kapazitäten an Stellen, die beispielsweise Puffer modellieren, ist dann nicht mehr sinnvoll möglich.

Die Petri-Netz-Modelle der Arbeitspläne beschreiben in GRAMAN die Ausführung von Fertigungsaufträgen (*orders*) auf verschiedenen Abstraktionsniveaus. Dieser Modellteil tauscht mit dem strukturellen Modell bei der Simulation Nachrichten über einen Rendezvous-ähnlichen Mechanismus aus. In der vorliegenden Arbeit wird statt dessen ein Gesamtmodell aus den beiden Beschreibungsformen erzeugt, welches dann untersucht werden kann. Da in GRAMAN „normale“ farbige Petri-Netze eingesetzt werden, sind viele Beschriftungen im Modell notwendig.



```

color Part = with WT.leer | A.Gel | A.Bel | A.Mont | A.Fert;
var v, x, y, z = Part;
fun FRob(x,y,z,v) = x=A.Gel and y=A.Bel and z=WT.leer and v=empty or
                  x=empty and y=WT.leer and z=A.Mont and v=A.Fert;
    
```

Abbildung 2.7: Farbiges Teilmodell des Roboters

In Bild 2.7 ist wie im vorherigen Abschnitt ein Teil des Beispiel-Fertigungssystems aus Kapitel 2.2 modelliert, hier aber mit einem farbigen Petri-Netz nach [40]. Das Modell enthält wieder nur den Roboter mit seinen Verbindungen und den Arbeitsplan für das Produkt A. Stellen haben neben ihrem Namen nun eine kursiv geschriebene Typbezeichnung. Die in runden Klammern stehenden Kapazitäten wurden als Erweiterung in [18] vorgeschlagen. Eine natürliche Zahl stellt dabei eine Kurzform für die ansonsten notwendige Angabe von *capacity colour set*, *capacity projection* und *capacity multi-set* dar.

Die Kanten sind mit Variablen oder konstanten Multimengen beschriftet. Variablen sind von einem bestimmten Typ, der genauso wie die Typen der Stellen unten in einem Textbereich deklariert werden müssen. Die dafür verwendete Sprache orientiert sich an der funktionalen Programmiersprache *standard ML*. Dabei sind sehr komplexe Ausdrücke möglich, die in diesem einfachen Beispiel nicht benötigt werden. Für die Modellierung von Systemen mit einem hohen Grad an Symmetrien bzw. mathematisch leicht zu beschreibenden Vorgängen eignet sich diese Methode sehr gut. Dies ist aber in Fertigungssystemen häufig nicht der Fall, was zu schlecht verständlichen Modellen führen kann. Im Beispiel müssen alle verschiedenen Be-

legungen der Transition **Roboter**, unter der sie schalten darf, in der Schaltfähigkeitsfunktion **FRob** angegeben werden.

In einem farbigen Petri-Netz sind alle Schaltmöglichkeiten einer Transition (*bindings*) konkurrierend zueinander schaltfähig [40]. Damit entsteht wieder das Problem der Modellierung exklusiver Betriebsmittel, die eine andere Schaltsemantik der Transitionen erfordern. Im dargestellten Beispiel wurde darauf verzichtet, dies mit zusätzlichen Transitionen und Stellen nachzubilden. In [88] wird ein größeres Beispiel mit einem farbigen Petri-Netz modelliert und mit dem entsprechenden farblosen Netz verglichen.

Ein Nachteil bei der Modellierung von Fertigungssystemen mit farbigen Petri-Netzen ist die Notwendigkeit, komplexe programmiersprachenähnliche Definitionen von Farben und Variablen anzugeben. Es gibt im Vergleich zu Stellen-Transitions-Netzen eine sehr viel größere Anzahl textueller Elemente im Modell, wodurch die intuitive Verständlichkeit des Modells stark erschwert wird. Darüber hinaus kann ein und dasselbe System auf sehr unterschiedliche Art und Weise abgebildet werden, was ein Modell als Basis für den Informationsaustausch zwischen den an der Entwicklung Beteiligten untauglich machen kann. Arbeitsplan und Struktur müssen von Hand in ein Modell integriert werden. Vorteilhaft ist dagegen die kompaktere Darstellung und die mögliche hierarchische Aufteilung des Modells in Teilmodelle.

2.5 Andere Beschreibungsmittel

Neben Petri-Netzen gibt es viele weitere Ansätze, Fertigungssysteme zu modellieren und zu untersuchen. Dabei werden vor allem Simulationssprachen und grafische Simulationswerkzeuge eingesetzt.

Universelle Simulationssprachen sind zwar nicht auf das spezielle Anwendungsgebiet der Fertigungssysteme zugeschnitten, bieten aber zumindest im Vergleich mit der Programmierung eines Simulators in einer normalen Programmiersprache vorgefertigte Elemente und Erleichterungen. Dazu gehört das Generieren von Zufallszahlen, die Verwaltung der Modellzeit und einer *event list* sowie das Sammeln und Aufbereiten von Daten zu Ergebnissen. Beispiele hierfür sind GPSS, SIMAN, SIMSCRIPT und SLAM [45]. Für einige davon sind auch Pakete mit Modulen für Fertigungssysteme erhältlich. Simulationssprachen schränken den Benutzer in seinen Möglichkeiten kaum ein, erfordern aber großen Programmieraufwand. Die entstehenden Programme sind schwer zu warten und für andere Entwickler wenig verständlich. Einige der genannten Sprachen besitzen neben der textuellen auch eine grafische Beschreibung. Visualisierung und Analyse sind aber anhand des Simulationsprogramms unmöglich oder müssen zusätzlich programmiert werden.

Als Beispiele für spezielle Simulationswerkzeuge für Fertigungssysteme seien AutoMod, ProModel, SIMFACTORY, WITNESS, XCELL+ und PERFACT! genannt. Meist handelt es sich dabei um Werkzeuge, die spezielle Konstrukte aus dem Fertigungsbereich sowie zur animierten Darstellung des Fertigungsablaufs enthalten. Dabei kann das Modell über eine menügesteuerte Oberfläche eingegeben werden, so daß kaum noch Programme eingegeben werden müssen. Schnittstellen zur Einbindung eigener Programme werden aber oft zusätzlich angeboten.

Das Softwarewerkzeug SimPro der inpro GmbH wird zur Modellierung und Simulation von Fertigungsabläufen verwendet. Das Grafische Basis-Sprache genannte Modellierungsmittel basiert auf Prädikat-Transitions-Netzen [11]. Jeder Transition sind Modula-2-Prozeduren zugeordnet, die bei ihrer Aktivierung ausgeführt werden und so die Eigenschaften der Transition und den Folgezustand bestimmen. Es werden hierarchische Modellierung sowie eine Bibliothek von Teilmodellen spezieller Anwendungsfälle unterstützt.

Eine bekannte Methode zur funktionalen Modellierung von Fertigungssystemen ist IDEF, die speziell für diesen Zweck aus SADT weiterentwickelt wurde [81]. Ein solches Modell besteht aus Funktionen und Objekten, die hierarchisch verfeinert werden können. Die Darstellung von Fertigungsvorgängen mit Funktionssymbolen wird in verschiedenen Arbeiten vorgeschlagen [66]. Neben dem weiter unten näher erläuterten System MOSYS verwenden sie auch das Beschreibungssystem MDM (*manufacturing description method*) der IBM Deutschland. Die Analyse und Auslegung von Fertigungssystemen mit Funktionssymbolen wird in der VDI-Richtlinie 3300 beschrieben.

Die in Kapitel 5.2 vorgeschlagene symbolische Beschreibung eines Fertigungssystems orientiert sich am Modellierungswerkzeug MOSYS [66, 55], das am Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik Berlin (IPK) entwickelt wurde. Es wird darum im folgenden näher beschrieben.

Das Modell eines Fertigungssystems besteht in MOSYS aus der funktionalen Beschreibung, den Arbeitsplänen und einer optionalen topologischen Beschreibung. Arbeitspläne werden als Tabellen dargestellt, die auch Alternativen enthalten können. Außerdem wird eine grafische funktionale Beschreibung des Fertigungssystems verwendet, die verschiedene Symbole für Tätigkeiten wie Fertigen, (De-)Montieren, Transportieren, Prüfen, Lagern sowie Systemein- und Ausgänge zur Verfügung stellt. Die einzelnen Symbole sind hierarchisch verfeinerbar. Zusätzlich kann eine topologische Beschreibung des Fertigungssystems gezeichnet werden, die den Grundriß mit der Anordnung aller Betriebsmittel und Transportwege beinhaltet. Sie dient zur Visualisierung des Modellverhaltens. MOSYS-Modelle können mit Simulation und Kostenrechnung bewertet werden, wofür Simula verwendet wird.

Die funktionale, symbolische Beschreibung modelliert nicht die Struktur des Fertigungssystems, sondern Materialfluß und funktionale Abfolge von Arbeitsgängen. Darum ist es mit dieser Beschreibungsmethode nicht ohne weiteres möglich, materialflußübergreifende Strukturen zu modellieren. Strukturelle Relationen von Betriebsmitteln sind im MOSYS-Modell versteckt. Die Verwendung von sogenannten Token, die durch Petri-Netze motiviert ist, ergänzt darum die Modellierungstechnik. Damit ist eine Art betriebsmittelorientierte Synchronisation zwischen Fertigungsabläufen möglich, die sich endliche Ressourcen teilen. Leider ist dies nicht Teil der grafischen Darstellung. Grundbausteine können durch Parameter an die anwenderspezifischen Anforderungen angepaßt werden. Die Menge und Bedeutung der Parameter ist durch die vielen zu beachtenden Einflußgrößen zum Teil für den Modellierer kaum noch überschaubar, da sie für viele spezifische Einzelprobleme erweitert wurden.

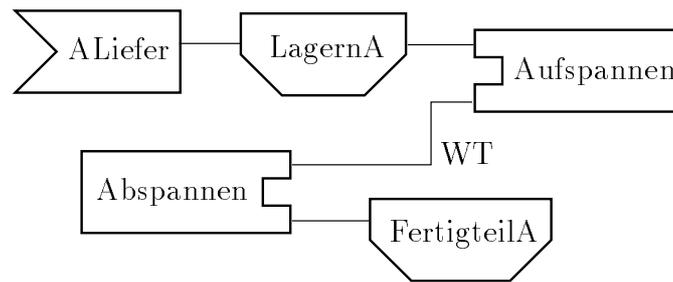


Abbildung 2.8: MOSYS-Teilmodell

Bild 2.8 zeigt wieder einen kleinen Ausschnitt aus dem fiktiven Beispiel, das in Abschnitt 2.2 beschrieben wurde. Die Arbeitsgänge des Roboters am Werkstück A sind als Aufspannen und Abspannen in der Be- und Entladestation modelliert. Da es sich hier um eine funktionale Beschreibung handelt, ist aus dem Modell nicht klar, daß es sich dabei um dasselbe Betriebsmittel handelt. Außerdem ist schlecht zwischen Roboter und Ladestation unterscheidbar, die ja eigentlich eine Pufferfunktion hat. Solche funktionsübergreifenden Einschränkungen werden durch Token abgebildet, die man in den Parametern der Symbole angibt. Das Modell kann dadurch mißverständlich werden, daß keine klare Trennung zwischen Betriebsmitteln und ihren Funktionen gezogen wird. Die Eigenschaften der Fertigungsarbeitsgänge sind sowohl in den Parametern der Funktionssymbole als auch in den tabellarischen Arbeitsplänen enthalten, die in der Darstellung weggelassen wurden.

Moderne Modellierungs- und Simulationswerkzeuge für Fertigungssysteme erfordern keinen oder kaum Programmieraufwand und sind daher auch ohne entsprechende Vorkenntnisse verwendbar. Oft sind aber für eine genauere Spezifikation des Verhaltens doch wieder programmähnliche Konstrukte notwendig, die meist einer eigens definierten Syntax entsprechen müssen. Die Stärke vieler verfügbarer Simulationswerkzeuge liegt in einer detailreichen Animation des Verhaltens des Modells, was einer der Gründe für ihre Popularität ist. Weniger Beachtung wird oft der statistisch abgesicherten Berechnung der Leistungsparameter und ihrer Konfidenzintervalle geschenkt. Aufgrund der Art der eingesetzten Modellarten ist die Untersuchung der Modelle auf Animation und Simulation eingeschränkt. Die hierarchische Verfeinerbarkeit wird nur von wenigen Werkzeugen unterstützt.

2.6 Einordnung der Arbeit

In dieser Arbeit wird eine auf Petri-Netzen basierende durchgängige Modellierungs- und Untersuchungsmethode für Fertigungssysteme vorgestellt. Dabei sollen die vorteilhaften Eigenschaften von Petri-Netzen für ein spezielles Anwendungsgebiet besser nutzbar gemacht werden.

Klassische Petri-Netze mit nicht unterscheidbaren Marken sind für die Modellierung nicht-trivialer Fertigungssysteme kaum verwendbar. Stellen und Transitionen müssen abhängig von den auftretenden Produktarten vervielfältigt werden. Dies führt zu großen und unübersichtlichen Modellen. Farbige Petri-Netze bieten die Möglichkeit, den Marken Attribute zuzuweisen und dadurch kompaktere Modelle zu erstellen. Außerdem sind größere Modelle

hierarchisch strukturiert. Aufgrund ihrer universellen Einsetzbarkeit ist jedoch die programmiersprachenartige Definition von Typen, Variablen und Funktionen notwendig. Diese und weitere textuelle Elemente erschweren die intuitive Verständlichkeit eines Modells. Es ist dem Modellierer überlassen, welche Modellinformationen er im grafischen bzw. textuellen Teil angibt. Der große Vorteil von Petri-Netzen, der in der grafisch verständlichen Beschreibung eines Systems liegt, wird dadurch für farbige und klassische Modelle relativiert.

Darum werden in dieser Arbeit für das Anwendungsgebiet der Fertigungssysteme spezielle farbige Petri-Netze eingeführt [88], in denen zwei Farbtypen vordefiniert sind und sich auch die grafischen Elemente an dieser Trennung orientieren. Dadurch werden viele der textuellen Elemente im Modell sowie der programmiersprachenartige Definitionsteil farbiger Netze überflüssig. Die Verwendung grafisch unterschiedlich markierter Netzobjekte wurde auch in [40] vorgeschlagen, aber nur zur Verbesserung der Lesbarkeit eines Modells und nicht zur Vermeidung von textuellen Definitionen. Der Modellierer ist durch die vorgegebenen Stellen- und Farbtypen gezwungen, das Modell entsprechend der realen Struktur zu entwerfen. Es ist dadurch leichter verständlich und kann für die Visualisierung des Verhaltens verwendet werden. Teilmodelle aus einer entsprechenden Bibliothek können nach ihrer Parametrisierung im Modell verwendet werden. Jeder Stelle ist eine Kapazitätsfunktion zugeordnet, mit der auch komplexe Sachverhalte modellierbar sind.

Mit den speziellen Petri-Netzen werden neben dem Strukturmodell auch davon getrennte Modelle der Arbeitspläne erstellt. Aus diesen Teilen wird automatisch ein Gesamtmodell erzeugt, das äußerlich dem Strukturmodell entspricht und dadurch gut verständlich ist. Dazu wurde das Konzept der Übergangstabellen entwickelt. Da aus den Arbeitsplanmodellen alle möglichen Belegungen ablesbar sind und daraus automatisch ein Gesamtmodell erzeugt wird, sind alle möglichen Belegungen der Transitionen (*bindings* nach [40]) bekannt. Sie sind darum explizit in den Übergangstabellen einer Transition enthalten. Die Definition und Verwendung von Variablen für die Beschreibung des Verhaltens einer Transition ist dann unnötig.

Die Analyse der entstehenden Modelle ist zum Teil einfacher als bei den *coloured Petri nets* nach Jensen möglich, da sich die Einschränkungen auf der Modellebene ausnutzen lassen. Um eine einfachere Modellierung der in einem Fertigungssystem vorherrschenden Betriebsmittel zu ermöglichen, wird eine spezielle Schaltsemantik von Transitionen eingeführt. Für die Leistungsbewertung sind stochastische Schaltzeiten mit den Transitionen verknüpft. Die numerische Analyse der Modelle kann auf der Ebene des Erreichbarkeitsgraphen mit denselben Methoden erfolgen, wie sie in [32] für klassische Petri-Netze mit nichtexponentiellen Schaltzeitverteilungen (*ϵ DSPNs*) entwickelt wurden. Alternativ wird die Berechnung von Leistungsparametern mit Hilfe von diskreter Ereignissimulation vorgenommen. Die Spezifikation dieser Parameter wird mit der Syntax vorgenommen, die auch sonst für Ausdrücke im Modell verwendet wird. Die beschriebenen Modellierungs- und Untersuchungstechniken wurden in einem Softwarewerkzeug realisiert.

Die Verwendung einer mathematisch fundierten Modellierungsmethode unterscheidet den beschriebenen Ansatz von gebräuchlichen Simulationswerkzeugen für Fertigungssysteme. Diese sind einer Untersuchung der Modellstruktur sowie der direkten numerischen Analyse nicht zugänglich.

Ein Modell eines Fertigungssystems als *coloured adaptive structured Petri net* wird nach [29] aus einer Menge von Regeln, Fakten, Prozeduren und weiteren Informationen in mehreren Zwischenschritten abgeleitet. Das entstehende Modell ist darum schwer durch den Modellierer interpretierbar. Es wird keine Trennung von Arbeitsplänen und Fertigungssystem vorgenommen. Mit den markierungsabhängigen Ausdrücken der hier verwendeten speziellen Petri-Netze können die Konstrukte dieser Netzklasse auf einfache Weise nachgebildet werden. Klassische Petri-Netze mit markierungsabhängigen Kantengewichten werden auch als *self-modifying nets* bezeichnet [21].

Im Unterschied zu GRAMAN werden in der vorliegenden Arbeit dieselben Modellarten für die Spezifikation der zwei Modellteile verwendet. Vertretertransitionen können durch Bibliotheksmodule verfeinert werden, deren strukturelle und andere Eigenschaften durch die Parametrisierung weitgehend beeinflussbar sind. Dazu enthält jedes Bibliotheksmodul außer den strukturellen Eigenschaften eine abstrakte Beschreibung der möglichen Arbeitsabläufe. Beide Teile eines Bibliotheksmoduls werden später parametrisiert und instanziiert, wenn sie in einem Modell verwendet werden.

Die als *scheduling* bezeichnete Bestimmung eines Zeitplans für die in einem Fertigungssystem auszuführenden Aufgaben ist ein komplexes Problem, für das bisher keine allgemein anwendbare Lösung gefunden wurde. Die vorliegende Arbeit beschäftigt sich nicht mit dieser Problemstellung, die z.B. in [36, 58, 78, 46, 65] im Zusammenhang mit Petri-Netzen bearbeitet wird. Vorhandene Steuerregeln können aber in das Modell integriert werden. Ein Modell eines Fertigungssystems beschreibt zu einem Zeitpunkt seinen Zustand (Markierung des Petri-Netzes) sowie die strukturell möglichen Aktivitäten (schaltfähige Transitionen). Eine Feinplanungsstrategie kann lediglich die möglichen Aktivitäten abhängig vom aktuellen Zustand einschränken. Dies ist in einem speziellen Petri-Netz leicht über die Angabe von Schaltbedingungen an Transitionen möglich. Eine solche Schaltbedingung ist ein möglicherweise komplexer markierungsabhängiger boolescher Ausdruck, der in einer Markierung wahr sein muß, damit die Transition schalten darf. Strategien wie z.B. „wenn Maschine A belegt oder ausgefallen, transportiere Teil X zu Maschine B und bearbeite es dort“ sind damit ohne weiteres möglich. Die Überwachung eines festen Zeitplanes kann dagegen nicht modelliert werden, da bei der Leistungsbewertung im eingeschwungenen Zustand keine abfragbare Zeit vorhanden ist. Bei einer entsprechenden Erweiterung wäre der Zustandsraum des Modells unendlich groß, weil der Zustandsbegriff oder die Marken dann um einen Zeitpunkt ergänzt werden müßten, der beliebige positive Werte annehmen kann. Für die transiente Analyse bzw. Simulation wäre dies prinzipiell denkbar, wurde bisher aber nicht weiter verfolgt.

Diese Dissertation beschäftigt sich ebenfalls nicht mit weiteren Problemen der Produktionsplanung und -steuerung wie Kostenrechnung, Bedarfsplanung, Losgrößenbestimmung, Fertigungssteuerung, Layoutplanung und anderen. Wenn aber der Einfluß entsprechender Festlegungen und Strategien in ein Modell integriert werden, sind ihre Auswirkungen auf die Leistungsfähigkeit und Zuverlässigkeit des geplanten Fertigungssystems indirekt untersuchbar. Die beste gefundene Lösung kann dann ausgewählt werden.

Kapitel 3

Modellierung von Fertigungssystemen mit speziellen Petri-Netzen

Dieses Kapitel beschreibt die in der vorliegenden Arbeit entwickelte Modellierungsmethode. Im ersten Abschnitt wird eine anwendungsspezifische Netzart definiert und erläutert, die von farbigen Petri-Netzen abgeleitet wurde. Dabei ist es durch die Beschränkung der verwendbaren Typen (Farben) von Marken nicht mehr notwendig, umfangreiche textuelle Beschriftungen und programmähnliche Konstrukte anzugeben.

Für komplexe technische Systeme wie Fertigungssysteme ist die gemeinsame Darstellung aller notwendigen Informationen ab einer gewissen Größe nicht mehr möglich. Dementsprechend werden Techniken zur Abstraktion und Verfeinerung benötigt, die durch die hierarchische Modellierung eröffnet werden. Vertretertransitionen stehen dabei für Unterseiten eines Modells, also Beschreibungen auf einer tieferen Ebene. Der Modellierer muß sich nur mit den auf der jeweiligen Abstraktionsstufe relevanten Informationen beschäftigen.

Um oft auftretende Modellteile nicht immer wieder neu spezifizieren zu müssen, ist die Verwendung einer Modulbibliothek sinnvoll. Einander ähnliche Module, die sich zum Beispiel nur durch Parameterwerte unterscheiden, können aus demselben Grundmodell abgeleitet werden. Dazu müssen die in der Bibliothek enthaltenen Teilmodelle mit den gewünschten Parametern instanziiert und in das Modell eingesetzt werden. Wie dies innerhalb der hier vorgestellten Modellierungsmethode geschieht, erläutert Abschnitt 3.3 näher.

Bei der Modellierung eines Fertigungssystems ist es notwendig, die Arbeitspläne der zu produzierenden Werkstücke mit in das Modell einzubeziehen. Abschnitt 3.2 zeigt, wie mit der hier vorgeschlagenen Methode die getrennte Modellierung von Fertigungssystem und Arbeitsplänen mit Petri-Netzen möglich ist. Damit wird eine klare Trennung zwischen Struktur und Funktion des Fertigungssystems eingeführt, wobei für beide Modellteile die im Abschnitt 3.1.1 definierten Petri-Netze verwendet werden.

Abschnitt 3.4 erläutert, wie dann die automatische Erzeugung eines Gesamtmodells aus den Beschreibungen der Struktur und der Arbeitsgänge erfolgt. Dieses Gesamtmodell beschreibt die Abläufe im Fertigungssystem bei der Produktion der in den Arbeitsplänen angegebenen Werkstücke und kann für die nachfolgende Analyse oder Simulation verwendet werden. Mit Untersuchungsmethoden für die entstehenden Modelle beschäftigt sich das Kapitel 4.

3.1 Eine Petri-Netz-Klasse für Fertigungssysteme

Dieser Abschnitt führt die verwendete Art von speziellen Petri-Netzen ein. In den folgenden Unterabschnitten wird die grundlegende Netzklasse definiert und Erläuterungen dazu gegeben. Um nicht für alle verwendeten Netzarten jeweils eine Definition angeben zu müssen, wird nur die zentrale Netzart, an der die spätere Untersuchung stattfindet, vollständig definiert. Für andere Modellarten werden leicht abgewandelte Versionen davon verwendet, wobei jeweils nur die Unterschiede dargestellt werden. Dies geschieht für das Strukturmodell des Fertigungssystems sowie das Arbeitsplanmodell in Abschnitt 3.2 und für die generischen Modelle der Bibliothek im Abschnitt 3.3.

Für die definierte Netzklasse werden die erlaubten Farbtypen der farbigen Petri-Netze [40] eingeschränkt. Als Farben bzw. Typen von Marken werden nur zwei Arten zugelassen: ungefärbte, ununterscheidbare Marken (Elementarmarken) für logische Zustände und die Farbe „Produkt“. Marken der Farbe „Produkt“ (Produktmarken) entsprechen z.B. einem Werkstück oder einem Werkstückträger im Fertigungssystem. Damit kann der Materialfluß getrennt von den Zustandsübergängen der Betriebsmittel modelliert werden.

3.1.1 Definition

Im folgenden wird eine Klasse farbiger, hierarchischer Petri-Netze definiert, die speziell für die Modellierung von Fertigungssystemen entwickelt wurde. Vorarbeiten dazu sind in [88, 89, 86] enthalten. Dabei werden Multimengen (*multi sets*) verwendet, die im Anhang B näher erläutert sind. Die hier vorgestellte Definition bildet den Grundtyp der verwendeten Modellklasse. Alle verwendeten Symbole sind im Anhang ab Seite 142 in einer Übersicht zusammengefaßt.

Ein **spezielles farbiges Petri-Netz** (sCPN) ist ein 10-Tupel:

$$\text{sCPN} = (\Sigma, S, C, K, I, G, U, SS, SZ, AZ),$$

dessen Elemente im folgenden näher erläutert werden.

- Die Menge der **Farbtypen** $\Sigma = \{E, \Phi\}$ besteht aus den Mengen E und Φ , in denen alle im jeweiligen farbigen Modell zugelassenen Markenarten (Farben) enthalten sind. Dabei ist
 - E die einelementige Menge der Elementarfarben: $E = \{e\}$. Marken dieses Typs sind ununterscheidbar wie Marken in farblosen Petri-Netzen. Sie können deshalb neben der Schreibweise als Multimenge (z.B. 3'e) auch grafisch als gefüllte Kreise „•“ dargestellt werden.
 - Φ die Menge der Markentypen, die Werkstücke oder andere physische Objekte sowie ihre Zustände im Fertigungssystem beschreiben. Diese Farbe ist darum aus drei Teilen zusammengesetzt: $\Phi = W \times V \times B$, und zwar aus der Menge der Werkstücke W , den Bearbeitungsstufen B der Teile und den verschiedenen Versionen V zur Unterscheidung alternativer Wege in einem Arbeitsplan. Die Mengen

W , V und B enthalten die für das jeweilige Modell notwendigen Elemente. Beispiele für Marken dieser Farbe sind z.B. **Rad[1]fertig**. Diese Farbmenge wird auch als zusammengesetzte oder Produktfarbe bezeichnet.

- Ein sCPN ist wie ein farbiges Petri-Netz in **Seiten** (*pages*) s aus der **Seitenmenge** S unterteilt. Dadurch wird eine hierarchische Aufteilung des Netzes möglich, und es entsteht eine baumartige Struktur der Seiten. Die **Primärseite** (*prime page*) s_p ist als oberste Beschreibungsebene die Wurzel dieses Baumes. Alle anderen Seiten befinden sich in der Hierarchie direkt oder indirekt unter ihr und sind jeweils über eine Vertretertransition (s.u.) mit der direkt übergeordneten Seite verbunden. Den Seiten sind die Elemente des Netzes zugeordnet:

$$\forall s \in S : s = (T_s, P_s, A_s)$$

Die Zuordnung der Elemente zu den Seiten ist eindeutig:

$$\forall s_1, s_2 \in S : s_1 \neq s_2 \rightarrow (T_{s_1} \cup P_{s_1} \cup A_{s_1}) \cap (T_{s_2} \cup P_{s_2} \cup A_{s_2}) = \emptyset$$

Die Elemente einer Seite sind im folgenden erklärt:

- T_s ist die Menge der **Transitionen** der Seite s . Die Menge T der Transitionen des gesamten Netzes ist definiert als $T = \bigcup_{s \in S} T_s$. Transitionen werden als Rechtecke dargestellt. Dabei können zur Verdeutlichung der verschiedenen Arten von Transitionen unterschiedliche Formen verwendet werden. Zeitbehaftete exponentielle Transitionen werden als ungefüllte Rechtecke dargestellt, deterministische Transitionen als gefüllte Rechtecke, zeitlose als dünne gefüllte Rechtecke und Vertretertransitionen als ungefüllte Rechtecke mit breiten oberen und unteren Streifen.
- P_s sind die **Stellen** (*places*) der Seite s . Die Menge P der Stellen des gesamten Netzes ist definiert als $P = \bigcup_{s \in S} P_s$. Die Mengen der Stellen und Transitionen eines Netzes ist disjunkt, d.h. $P \cap T = \emptyset$. Als Menge der **Elementarstellen** P^E wird die Menge der Stellen des Modells bezeichnet, die elementare Marken enthalten können¹: $P^E = \{p \in P \mid C(p) = E\}$. Elementarstellen werden als dünne Kreise gezeichnet. Entsprechend ist die Menge der **Produktstellen** P^Φ die Menge der Stellen, die Produktmarken enthalten können: $P^\Phi = \{p \in P \mid C(p) = \Phi\}$. Produktstellen werden als dicke Kreise gezeichnet.
- Die **Kanten** (*arcs*) A_s sind gerichtete Verbindungen zwischen den Transitionen und Stellen einer Seite s : $A_s \subseteq T_s \times P_s \cup P_s \times T_s$. Sie werden als Pfeil in der entsprechenden Richtung dargestellt. Die Menge A der Kanten des gesamten Modells ist wie folgt definiert: $A = \bigcup_{s \in S} A_s$. Als Menge der **Elementarkanten** A^E wird die Menge der Kanten bezeichnet, die mit einer Elementarstelle verbunden sind: $A^E = \{(p, t), (t, p) \mid p \in P^E\}$. Elementarkanten werden dünn gezeichnet. Entsprechend ist die Menge der **Produktkanten** A^Φ die Menge der Kanten, die mit einer Produktstelle verbunden sind: $A^\Phi = \{(p, t), (t, p) \mid p \in P^\Phi\}$. Produktkanten werden dick dargestellt. Eine Kante a wird genau dann als Eingangskante einer Transition $t \in T$ und Ausgangskante einer Stelle $p \in P$ bezeichnet, wenn

¹Die Farbfunktion C wird weiter unten definiert.

gilt: $a = (p, t)$. Entsprechend wird eine Kante a genau dann als Ausgangskante einer Transition $t \in T$ und als Eingangskante einer Stelle $p \in P$ bezeichnet, wenn $a = (t, p)$. Die Menge der **Eingangsstellen** P^{in} bzw. **Ausgangsstellen** P^{out} einer Transition t ist die Menge der Stellen, die mit der Transition über Eingangs- bzw. Ausgangskanten verbunden sind:

$$P^{in}(t) = \{p \in P \mid \exists (p, t) \in A\}$$

$$P^{out}(t) = \{p \in P \mid \exists (t, p) \in A\}$$

- Die **Farbfunktion** C bildet Stellen und Kanten auf die jeweils zugehörige Farbmenge ab: $C : P \cup A \rightarrow \Sigma$. Sie gibt für eine Stelle an, welche Art Marken sich in ihr befinden dürfen. Dementsprechend werden Elementar- und Produktstellen unterschieden (s.o.). Für Kanten ist die Farbfunktion implizit durch die Farbfunktion der mit der Kante verbundenen Stelle gegeben: $\forall a \in A, \forall p \in P : a = (t, p) \vee a = (p, t) \rightarrow C(a) = C(p)$
- Die **Kapazitätsfunktion** $K : P \times R \rightarrow \mathbf{IB}$ spezifiziert das maximale Markenfassungsvermögen einer Stelle². Sie wird in eckigen Klammern nahe der Stelle dargestellt. Um komplexe Kapazitätsbeschränkungen angeben zu können, ist die Kapazität einer Stelle ein boolescher markierungsabhängiger Ausdruck. Ist der Wert dieses Ausdrucks in einer bestimmten Markierung Wahr, ist die Kapazität der Stelle nicht überschritten (und umgekehrt). Wird keine Kapazität angegeben, hat die Stelle ein unbegrenztes Fassungsvermögen; der Kapazitätsausdruck hat also den Standardwert Wahr. Häufig wird der einfache Fall benötigt, in dem lediglich die Anzahl beliebiger Marken in einer Stelle begrenzt ist. Zur Vereinfachung der Schreibweise ist dafür alternativ die Angabe einer natürlichen Zahl als Kapazität möglich. Diese gibt die maximale Anzahl der Marken an, die sich in dieser Stelle zu einem Zeitpunkt befinden können. Die Angabe einer Zahl n als Kapazität einer Stelle P steht für den Kapazitätsausdruck $[\#P \leq n]$ an der Stelle³.
- Die **Initialisierungsfunktion** I bildet jede Stelle auf ihre Anfangsmarkierung ab. Damit ist die initiale Markierung r_0 des gesamten Netzes festgelegt (siehe Abschnitt 4.3.1). Der Standardwert ist die leere Multimenge⁴.

$$I : \begin{cases} P^E \rightarrow \mathcal{M}_E \\ P^\Phi \rightarrow \mathcal{M}_\Phi \end{cases}$$

- Die **Schaltbedingung** (*guard expression*) $G : T^N \times R \rightarrow \mathbf{IB}$ ist eine Funktion, die einer Transition in einer Markierung einen Wahrheitswert zuweist (ein markierungsabhängiger boolescher Ausdruck)⁵. Dieser Ausdruck wird in eckigen Klammern an der Transition dargestellt. Die Transition kann in einer Markierung nur dann aktiviert sein, wenn zusätzlich zu den sonstigen Bedingungen dieser Ausdruck *Wahr* als Ergebnis liefert.

² R ist die Menge der Markierungen und wird im Abschnitt 4.2 definiert.

³ $\#P$ steht für die Anzahl der Marken in der Stelle P , Erläuterung weiter unten.

⁴ \mathcal{M}_A ist die Menge aller Multimengen, die über der Menge A gebildet werden kann.

⁵ T^N bezeichnet die Menge der Transitionen, die keine Vertretertransitionen sind.

- Die **Übergangsfunktion** $U : T^N \rightarrow \{u_1, u_2, \dots, u_n\}$ bildet Transitionen auf eine Menge von Übergängen ab. Jeder einzelne **Übergang** u_i beschreibt dabei eine mögliche Art des Schaltens dieser Transition in bestimmten Markierungen:

$$u_i = (u_i^{guard}, u_i^{in}, u_i^{out}, u_i^{select}, u_i^{weight}, u_i^{delay})$$

Dazu werden die notwendigen logischen Bedingungen, die beteiligten physischen Objekte bzw. logische Zustände und deren Bewegungen oder Umwandlungen sowie das zeitliche Verhalten eines einzelnen diskreten Vorgangs im Fertigungssystem angegeben. Die Menge der in einer Markierung schaltfähigen Übergänge entspricht in ihrer Funktion der Menge der *bindings* einer Transition [40] in dieser Markierung. Die Übergangstabelle wird normalerweise in der grafischen Darstellung weggelassen. Dabei bedeutet innerhalb eines Übergangs

- $u^{guard} : R \rightarrow \mathbf{B}$ heißt Schaltbedingung des Übergangs. Dabei handelt es sich um eine Funktion, die für jede Markierung $r \in R$ einen Wahrheitswert liefert (ein markierungsabhängiger boolescher Ausdruck). Der Übergang ist in einer Markierung nur aktiviert, wenn dieser Ausdruck *Wahr* als Ergebnis liefert.
- u^{in} beschreibt, welche Marken durch das Schalten der Transition t unter diesem Übergang von den Eingangsstellen abgezogen (verbraucht) werden. u^{in} ist eine möglicherweise markierungsabhängige Abbildung von den Eingangsstellen der Transition auf Multimengen, wobei die Kanten in der Übergangstabelle durch die Angabe der verbundenen Stelle und ihre Richtung bezeichnet werden. Die entsprechende Syntax ist im Anhang C enthalten.

$$u^{in} : \begin{cases} (P^{in}(t) \cap P^E) \times R \rightarrow \mathcal{M}_E \\ (P^{in}(t) \cap P^\Phi) \times R \rightarrow \mathcal{M}_\Phi \end{cases}$$

- u^{out} beschreibt, welche Marken durch das Schalten der Transition t unter diesem Übergang zu den Ausgangsstellen bewegt (dort erzeugt) werden. u^{out} ist eine möglicherweise markierungsabhängige Abbildung von den Ausgangsstellen der Transition auf Multimengen.

$$u^{out} : \begin{cases} (P^{out}(t) \cap P^E) \times R \rightarrow \mathcal{M}_E \\ (P^{out}(t) \cap P^\Phi) \times R \rightarrow \mathcal{M}_\Phi \end{cases}$$

- $u^{select} \in \mathbf{R}$ ist eine positive reelle Zahl, die bei zeitbehafteten Transitionen mit der Schaltsemantik *exclusive server* (siehe 4.3.1) die relative Wahrscheinlichkeit dafür angibt, daß dieser Übergang aus der Menge der schaltfähigen Übergänge der Transition ausgewählt wird. Für alle anderen Transitionen kann er weggelassen werden.
- $u^{weight} \in \mathbf{R}$ ist eine positive reelle Zahl, die die relative Wahrscheinlichkeit für das Schalten des Übergangs angibt, falls mehrere Übergänge zum selben Zeitpunkt zu schalten versuchen.
- Die Schaltzeitverteilung $u^{delay} : R \rightarrow \mathcal{F}$ ist eine Wahrscheinlichkeitsdichtefunktion, die die Zufallsvariable der Schaltzeit des Übergangs in Abhängigkeit von

der aktuellen Markierung $r \in R$ beschreibt. u^{weight} und u^{delay} sind entsprechend der in [32] vorgeschlagenen Erweiterung für farblose stochastische Petri-Netze gewählt. \mathcal{F} entspricht der Menge der zugelassenen Schaltzeitverteilungen über dem Intervall $[0, \infty)$, die jeweils Dichtefunktionen angeben. An dieser Stelle sind verschiedene Formen zulässig, die in der Syntaxbeschreibung näher spezifiziert sind (siehe Anhang C). Die Klasse der zugelassenen Schaltzeitverteilungen wird als *expolynomial* bezeichnet, nähere Erläuterungen dazu sind im Abschnitt 4.3.2 zu finden.

Als zeitlos wird ein Übergang bezeichnet, für den $\forall r \in R, t \in \mathbb{R}^+ : u^{delay}(r)(t) = 0$ gilt; er kann dann nur zum Zeitpunkt $t = 0$ schalten. Alle anderen Übergänge werden als zeitbehaftet bezeichnet. Generell gilt: die Übergänge einer Transition müssen in allen Markierungen entweder alle zeitbehaftet oder alle zeitlos sein. Im ersten Fall handelt es sich bei der Transition um eine **zeitbehaftete**, im zweiten Fall um eine **zeitlose Transition**. Die Mengen der zeitbehafteten bzw. zeitlosen Transitionen werden mit T^{tim} (*timed transitions*) und T^{imm} (*immediate transitions*) bezeichnet. Die Schaltzeitverteilungen der Übergänge können diskrete und kontinuierliche Anteile haben. Aufgrund der einfachen Analysierbarkeit nimmt die Exponentialverteilung dabei eine Sonderstellung ein. Abhängig von den auftretenden Schaltzeitverteilungen ergeben sich verschiedene Klassen von mehr oder weniger kompliziert analysierbaren Modellen [20, 32], auf die im Kapitel 4.3.2 näher eingegangen wird.

- Die **Schaltsemantik** SS beschreibt die Art und Weise des Schaltverhaltens einer zeitbehafteten Transition näher.

$$SS : T^{tim} \rightarrow \{ES, IS\}$$

Bei einer Transition mit der Schaltsemantik *infinite server* (IS) können in einer Markierung alle Übergänge gemeinsam und konkurrierend schaltfähig sein. Im Normalfall (*exclusive server*, ES) kann die Transition nur unter einem Übergang schalten. Die Bedeutung wird im Abschnitt 4.3.1 näher erläutert. Transitionen mit der Schaltsemantik *infinite server* sind durch ein IS gekennzeichnet.

- Die **Seitenzuweisungsfunktion** ist eine eineindeutige Abbildung $SZ : T \rightarrow S$, die jeder Vertretertransition die zugehörige Seite zuweist, mit der sie verfeinert wird. Eine Vertretertransition ist ein Platzhalter für eine in der Hierarchie darunterliegende Seite. Jede Seite ist genau einer Vertretertransition zugeordnet, wenn sie nicht die Primärseite s_p ist: $\forall s \in S : \exists(t, s) \in SZ \vee s = s_p$. Die Menge der Vertretertransitionen T^V ist die Menge der Transitionen, denen eine Unterseite zugeordnet ist: $T^V = \{t \in T \mid \exists(t, s) \in SZ\}$. Die Menge der Transitionen, die nicht Vertretertransitionen sind, wird mit T^N bezeichnet: $T^N = T \setminus T^V$. Eine Seite s_1 heißt **Unterseite** einer anderen Seite s_2 genau dann, wenn sie einer Vertretertransition zugeordnet ist, die Teil der anderen Seite ist: $\exists t \in T_{s_2} : SZ(t) = s_1$. Dann heißt s_2 außerdem **Oberseite** von s_1 . Ausgehend von der Primärseite s_p wird so eine baumartige hierarchische Struktur gebildet. Dabei dürfen keine Zyklen auftreten.

- Der Austausch von Marken zwischen Seiten in der Hierarchie findet über Anschluß- und Sockelstellen statt. **Sockelstellen** sind alle die Stellen, die durch Ein- oder Ausgangskanten mit Vertretertransitionen verbunden sind. Als **Anschlußstellen** wird die Menge der Stellen einer Unterseite bezeichnet, über die auf die Sockelstellen der Oberseite zugegriffen werden kann. Jede Sockelstelle ist auf der Unterseite über eine Anschlußstelle sichtbar. Anschlußstellen werden auf der Unterseite gestrichelt dargestellt. Die **Anschlußzuweisung** $AZ : P \rightarrow P$ weist dazu jeder Anschlußstelle genau eine Sockelstelle auf der darüberliegenden Seite zu. Die zugewiesene Sockelstelle muß über eine Kante mit der Vertretertransition verbunden sein, der die Unterseite zugeordnet ist. Die Richtung der Kante zwischen der Vertretertransition und der Sockelstelle muß der Richtung der Kante(n) entsprechen, die auf der Unterseite mit der Anschlußstelle verbunden sind. Alle Attribute einer Anschlußstelle (Farb- und Kapazitätsfunktion) müssen stets mit den entsprechenden Attributen der zugeordneten Sockelstelle übereinstimmen:

$$\forall (p_i, p_k) \in AZ : C(p_i) = C(p_k) \wedge I(p_i) = I(p_k)$$

Eine Anschlußstelle läßt sich deshalb auch als Verweis auf die entsprechende Sockelstelle auffassen. Wenn die Anschlußzuweisung AZ die genannten Bedingungen erfüllt, läßt sich die Menge der Anschlußstellen P^A definieren: $P^A = \{p_1 \in P \mid \exists (p_1, p_2) \in AZ\}$. Die Menge der Stellen, die nicht Anschlußstellen sind, wird dann mit P^N bezeichnet: $P^N = P \setminus P^A$.

3.1.2 Erläuterungen

In diesem Abschnitt werden die Elemente der eben definierten Art von Petri-Netzen erläutert. Dabei wird auf ihre Verwendung und grafische Darstellung eingegangen.

Stellen

Elementarstellen $P \in P^E$ können nur elementare Marken aufnehmen und werden wie in klassischen Petri-Netzen durch dünne Kreise dargestellt. Mit ihnen können z.B. das Ausfallverhalten von Betriebsmitteln, die Verfügbarkeit von Hilfsstoffen sowie ganz allgemein Zustände von Ressourcen abgebildet werden. Produktstellen $P \in P^\Phi$ sind dementsprechend den Produktmarken vorbehalten und werden zur sichtbaren Unterscheidung als dicke Kreise gezeichnet. Sie repräsentieren Orte, an denen sich Objekte im Fertigungsprozeß befinden können. Durch die unterschiedliche grafische Darstellung sind viele der sonst in farbigen Petri-Netzen notwendigen textuellen Elemente im Modell überflüssig. Außerdem ist der Modellierer gezwungen, die Struktur des Fertigungssystems entsprechend der Realität zu modellieren, d.h. eine Produktstelle im Modell entspricht einem Ort, an dem sich Werkstücke befinden können. Dementsprechend sollten nie mit verschiedenen Produktstellen Werkstücke verschiedener Zustände am selben Ort modelliert werden, wie dies oft in der Literatur getan wird. Diese klare Trennung von Ort und Zustand der modellierten Objekte hilft, ein einfaches und übersichtliches Modell zu erzeugen.

Produktstellen $P \in P^\Phi$ modellieren wie oben beschrieben mögliche Aufenthaltsorte von Werkstücken und anderen physischen Objekten. Dazu gehören Lager, Puffer, Bearbeitungsplätze und ähnliche. Die Anzahl der in solch einem Ort befindlichen Werkstücke ist im allgemeinen physikalisch oder konstruktiv begrenzt. Diese Begrenzung ist ein wichtiges Attribut für die Beschreibung einer solchen Stelle. In der vorgeschlagenen Art Petri-Netze ist darum die Angabe einer Kapazität für jede Stelle möglich; sie wird an der Stelle in eckigen Klammern angegeben. Ist keine spezifiziert, wird sie als unbegrenzt angenommen. Dabei sind zwei Formen zugelassen: Im einfachen Fall eine natürliche Zahl, die die maximale Anzahl der Marken irgendeiner Art in der Stelle angibt. Hängt die Kapazität einer Stelle dagegen von der Art der Marken ab, z.B. weil die Werkstücke unterschiedlich groß sind, reicht dies nicht aus. Für diesen Fall darf als Kapazität ein beliebiger markierungsabhängiger boolescher Ausdruck angegeben werden. Ist das Ergebnis der Auswertung dieses Ausdrucks in einer bestimmten Markierung Falsch, ist die Kapazität der Stelle überschritten und die zu diesem Zustand führende Aktivität darf nicht ausgeführt werden. Passen in ein Lager zum Beispiel 4 Teile vom Typ A oder aber für jedes Teil A zwei Teile B, kann die Kapazität mit $[\#Lager(A) + 0.5*\#Lager(B) \leq 4]$ ausgedrückt werden. Die Syntax dieser Ausdrücke wird weiter unten erklärt, $\#Lager(A)$ entspricht der Anzahl Werkstücke vom Typ A in der Stelle *Lager*.

Zusätzlich zu den Informationen über den Aufbau und die Funktion eines Fertigungssystems muß für eine Untersuchung noch der Anfangszustand bekannt sein. Dazu kann etwa die Ausgangssituation bei Aufnahme der Produktion zu Beginn einer Schicht verwendet werden. Zu diesem Zweck kann für jede Stelle die Initialisierungsfunktion I angegeben werden. Dabei handelt es sich um einen Ausdruck, dessen Ergebnis den Inhalt der jeweiligen Stelle am Beginn der Auswertung angibt. Das Ergebnis des Ausdrucks muß dabei eine Multimenge von Farben sein, die dem Typ der Stelle entspricht. Wird kein Ausdruck angegeben, ist die Stelle zunächst leer. Die Initialisierungsfunktion kann grafisch wie jede andere Markierung dargestellt werden: Die Anzahl der Marken in einer Stelle kann zunächst als Zahl (oder bei geringen Mengen als einzelne schwarze Punkte) in der Stelle angegeben werden. Für Produktstellen muß mit einer entsprechenden Multimenge angegeben werden, welche Farben (Typen) die Marken haben.

Transitionen

Transitionen $t \in T$ modellieren Ereignisse bzw. Aktivitäten, die den Zustand des Modells verändern. Sie können dabei Marken von ihren Eingangsstellen abziehen und (möglicherweise andere) Marken auf ihren Ausgangsstellen ablegen. Dies ist wegen der Verwendung farbiger Netze nur eine vereinfachte Darstellung, denn die eigentlichen Aktivitäten werden durch die in den Transitionen enthaltenen Übergänge beschrieben. Zeitlose Transitionen schalten ohne Zeitverzögerung und werden als dünne gefüllte Rechtecke dargestellt. Wird dagegen eine zeitbehaftete Transition schaltfähig, muß zunächst eine von der Schaltzeitverteilung abhängige Zeit vergehen, bevor die Transition schaltet. Diese Transitionen werden als dicke Rechtecke dargestellt, wobei die Art der Schaltzeitverteilung ebenfalls grafisch angezeigt werden kann (exponentielle Schaltzeitverteilung: leeres Rechteck, deterministische Schaltzeit: gefülltes Rechteck). Aus den durch die Transitionen beschriebenen lokalen Zustandsänderungen im Modell ergibt sich das Verhalten des Modells, das einen stochastischen Prozeß beschreibt.

An einer Transition wird außer dem Namen die Schaltbedingung in eckigen Klammern dargestellt. Dabei handelt es sich um einen booleschen Ausdruck, der in einer Markierung wahr sein muß, damit die Transition schalten darf. Damit lassen sich notwendige Bedingungen für die Aktivierung eines Betriebsmittels angeben, wie zum Beispiel die Funktionsfähigkeit einer Maschine. Der Standardwert der Schaltbedingung ist Wahr, so daß in der grafischen Darstellung nur davon abweichende Bedingungen angegeben werden müssen. Im Kapitel 4.3.1 wird die Bedeutung der Schaltsemantik einer zeitbehafteten Transition erläutert.

Die Übergangstabelle einer Transition besteht aus einer Reihe von Übergängen u_i , die zusammen alle möglichen Verhaltensweisen der Transition beschreiben. Dazu enthält jeder Übergang unter anderem Angaben über die Schaltbedingung, die verbrauchten und erzeugten Marken und die Schaltzeitverteilung. Die automatische Erzeugung der Übergangstabellen wird im Abschnitt 3.4 erklärt. Der Anhang enthält auf Seite 145 eine Beschreibung des Formats einer Übergangstabelle.

Die Schaltbedingung des Übergangs u^{guard} ist ein boolescher Ausdruck, der in einer Markierung Wahr sein muß, damit der Übergang aktiviert werden kann. Das Eingangsverhalten u^{in} beschreibt, welche Marken beim Schalten des Übergangs von den Eingangsstellen abgezogen werden. Das Ausgangsverhalten u^{out} gibt dementsprechend an, welche Marken auf den Ausgangsstellen erzeugt werden. Das Ein- und Ausgangsverhalten spezifiziert also, welche physischen Objekte bzw. Zustände an dem durch den Übergang modellierten diskreten Vorgang beteiligt sind. Die Übergänge von zeitbehafteten Transitionen verbrauchen Zeit, bevor sie schalten. u^{delay} gibt die Verteilungsfunktion dieser Schaltzeit an, die markierungsabhängig sein kann. Im Anhang ist angegeben, welche Möglichkeiten zur Spezifikation dafür vorhanden sind. Wenn zu einem Zeitpunkt im Modell mehrere Übergänge schalten, muß eine Schaltreihenfolge bestimmt werden. u^{weight} gibt die Wahrscheinlichkeit dafür an, daß der Übergang als erster ausgewählt wird. Diese Wahrscheinlichkeit ist besonders für konkurrierende zeitlose Transitionen wichtig, da ihr Schalten keine Zeit verbraucht und sie demnach immer zum selben Zeitpunkt zu schalten versuchen. Zeitbehaftete Transitionen, die exklusiv nutzbare Betriebsmittel beschreiben, können mit der Schaltsemantik *exclusive server* abgebildet werden. Wird eine solche Transition schaltfähig, wird nur einer der theoretisch schaltfähigen Übergänge der Transition aktiviert. u^{select} gibt die relative Wahrscheinlichkeit dafür an, daß der Übergang unter den möglichen ausgewählt wird.

Kanten

Bei den hier verwendeten Petri-Netzen sind ausschließlich Kanten zugelassen, auf denen sich tatsächlich Marken „bewegen“ können, also keine hemmenden oder Testkanten. Dies erhöht die Verständlichkeit der Modelle, denn nur Bewegungen von Objekten bzw. Zustandsänderungen werden grafisch dargestellt. Andere Kanten sind nicht notwendig, da logische Bedingungen ohne Einschränkung der Modellierungsmächtigkeit in Schaltbedingungen und Kapazitäten angegeben werden können. Die verwendeten Kanten verbinden jeweils genau eine Transition mit einer Stelle und werden als Linien mit einem Pfeil am Ende dargestellt. Entsprechend der durch den Pfeil angegebenen Richtung einer Kante spricht man von Ein- und Ausgangskanten der Stellen bzw. Transitionen. Jede Kante ist mit genau einer Stelle verbunden, die entweder nur elementare oder Produktmarken enthalten kann. Durch die mit der Stelle verbundenen Kanten können demnach nur Marken der gleichen Farbe

bewegt werden. Der Typ der Kante entspricht also immer dem verbundenen Stellentyp, und dementsprechend werden Elementar- und Produktkanten unterschieden. Diese werden wieder wie die Stellen verschieden grafisch dargestellt: Elementarkanten sind dünn und Produktkanten dick gezeichnet. Die sichtbaren Kanten sind nur potentielle Kanten, denn die eigentliche Beschreibung des Schaltverhaltens der Transitionen steckt in den Übergängen. In den Übergängen einer Transition sind aber nur Zugriffe auf Stellen erlaubt, die mit der Transition über eine Kante des passenden Typs (Ein- oder Ausgang) verbunden sind.

Hierarchie

Die hierarchische Modellierung ermöglicht es, ein komplexes Modell aus Teilmodellen zusammenzufügen, die Teile der Beschreibung auf unterschiedlichen Abstraktionsstufen enthalten. Dazu sind die verwendeten Petri-Netze aus Seiten s aufgebaut. Ausgehend von der zentralen obersten Primärseite s_p können Seiten Vertretertransitionen \square enthalten, die als Platzhalter für Seiten auf der nächstniedrigeren Hierarchiestufe dienen. Jeder Vertretertransition ist über die Seitenzuweisungsfunktion SZ genau eine Unterseite zugeordnet. Durch die Zuweisung von Unterseiten zu Vertretertransitionen dürfen keine Zyklen in der Seitenstruktur entstehen. In Anlehnung an [40] wurden hier Transitionen für die Verfeinerung gewählt. Es wäre auch möglich, Stellen oder beide Objektarten für die Verfeinerung zuzulassen. Die Beschränkung auf Transitionen vereinfacht die Modellierungstechnik und verringert die Ausdrucksmöglichkeiten kaum. Vom Gesichtspunkt der Modellierung eines Fertigungssystems erscheinen Transitionen besser für die Darstellung der Verfeinerung eines Betriebsmittels geeignet, da Maschinen eine wichtigere Rolle als z.B. Lager spielen. Die Vertretertransitionen bzw. ihre Unterseiten können Marken enthalten, und sie haben kein echtes Schaltverhalten. Damit unterscheiden sie sich stark von den normalerweise im Petri-Netz auftretenden Transitionen.

Stellen, die mit einer Vertretertransition durch eine Kante verbunden sind, werden Sockelstellen genannt und stellen die Verbindung zwischen der darunterliegenden Seite und ihrer Umgebung auf der höheren Ebene her. Auf der Unterseite gibt es Anschlußstellen, die den Sockelstellen der Vertretertransition über die Anschlußzuweisung AZ eindeutig zugeordnet sind. Über die Anschlußstellen sind die Sockelstellen der Oberseite sichtbar. Sie werden gestrichelt gezeichnet und müssen in ihren Attributen der jeweiligen Sockelstelle entsprechen.

Ausdrücke und Schreibweisen

Wie bereits im vorangegangenen Kapitel beschrieben, besteht der Typ von Produktmarken Φ aus drei Teilen, und zwar dem Produkt W , den Bearbeitungsstufen B der Produkte sowie den verschiedenen Versionen V . Werkstücke und andere physische Objekte im Fertigungssystem, die durch eine Produktmarke modelliert werden, sind durch die Angabe des Produktes und der aktuellen Bearbeitungsstufe eindeutig beschrieben. Identische Produkte können aber für verschiedene alternative Fertigungsrouten vorgesehen sein. Darum muß eine Unterscheidung zwischen ansonsten gleichen Werkstücken möglich sein, die sich in unterschiedlichen Alternativen befinden. Dies geschieht in den hier verwendeten Petri-Netzen mit Hilfe der sogenannten Version des Werkstücks, einer natürlichen Zahl oder einem Namen. Normalerweise besitzt jedes Werkstück die Versionsnummer Null. An einer Stelle im

Fertigungssystem, an der eine Entscheidung über die Auswahl einer der möglichen alternativen Produktionsrouten getroffen wird, erhält jede von einer solchen Entscheidung betroffene Marke eine Versionsnummer ungleich Null, die die gewählte Route beschreibt. Dadurch kann später bei der Analyse des Modells lokal beim Schalten einer Transition auf die Information über die Fertigungsalternative zugegriffen werden.

Der Typ einer Produktmarke wird als Aneinanderreihung von Produkt, Version und Bearbeitungsstufe geschrieben. Die Versionsnummer steht dabei in eckigen Klammern, also z.B. **Gehäuse[0]entgratet**. Da der Standardfall des Werkstücks ohne besondere Versionsnummer oft auftritt, kann statt [0] auch ein Punkt geschrieben werden, also **Gehäuse.entgratet** für das Beispiel. Statt einer natürlichen Zahl kann die Version auch als Zeichenkette angegeben werden.

An verschiedenen Stellen treten in den speziellen Petri-Netz-Modellen Ausdrücke auf. Eine vollständige Darstellung der Syntax und der erlaubten Operationen ist im Anhang D enthalten. Das Ergebnis eines Ausdrucks kann eine Multimenge von Farben bzw. Marken, eine natürliche Zahl oder (durch die Verwendung von Vergleichen) ein Wahrheitswert sein. Die Ausdrücke können entweder fest oder markierungsabhängig sein. Ein markierungsabhängiger Ausdruck ist eine Funktion, deren Wert erst durch Auswertung der aktuellen Markierung des Modells bestimmt wird. Dazu stehen folgende Konstrukte zur Verfügung, zu deren Erklärung die Operationen auf Multimengen (Seite 144) und die Definition einer Markierung (Seite 70) verwendet werden.

- **@Platz** gibt den Inhalt der Stelle **Platz** in der aktuellen Markierung an. Dies entspricht dem Wert von $r_i(\text{Platz})$ in der Markierung r_i . Das Ergebnis des Ausdrucks ist eine Multimenge von Farben oder Marken aus der Farbmenge der Stelle.
- **@Platz(Auswahl)** gibt den Inhalt der Stelle **Platz** in einer Markierung an, wobei nur Marken mit Farben berücksichtigt werden, die in der Multimenge **Auswahl** enthalten sind. Dies entspricht dem Wert von $r_i(\text{Platz})/\text{Auswahl}$ in der Markierung r_i . Das Ergebnis des Ausdrucks ist ebenfalls eine Multimenge von Farben aus der Farbmenge der Stelle.
- **#Platz** gibt die Anzahl der Marken in Stelle **Platz** in einer Markierung an. Dies entspricht dem Wert von $|r_i(\text{Platz})|$ in der Markierung r_i . Das Ergebnis des Ausdrucks ist eine natürliche Zahl.
- **#Platz(Auswahl)** gibt die Anzahl der Marken in Stelle **Platz** in einer Markierung an, wobei nur Marken mit Farben berücksichtigt werden, die in der Multimenge **Auswahl** enthalten sind. Dies entspricht dem Wert von $|r_i(\text{Platz})/\text{Auswahl}|$ in der Markierung r_i . Das Ergebnis des Ausdrucks ist ebenfalls eine natürliche Zahl.

In den eben genannten Konstrukten zum Zugriff auf Stelleninhalte in Ausdrücken tritt eine **Auswahl** genannte Multimenge auf. Um beispielsweise die Anzahl aller Teile eines Produkttyps in einer Stelle abzufragen, müßte man die Menge aller auftretenden Bearbeitungsstufen und Varianten in **Auswahl** angeben. Zur Vereinfachung der Schreibweise dieses häufigen Falls werden folgende Jokerzeichen (*wildcards*) zusätzlich zugelassen:

- **Werkstück[?]**Zustand steht für alle Teile vom Typ **Werkstück** im Zustand **Zustand**. Die Variante ist beliebig.
- **Werkstück?** steht für alle Teile vom Typ **Werkstück**; der Bearbeitungszustand und die Variante sind beliebig.

Die Anzahl aller Teile vom Typ **A** in der Stelle **Platz** ergibt sich dann z.B. durch **#Platz(A?)**.

Um einmal definierte Ausdrücke oder Werte mehrfach verwenden zu können, werden **Definitionen** verwendet. Eine Definition besteht aus einem Namen und einem möglicherweise markierungsabhängigen Ausdruck, der den Wert der Definition bestimmt. Sie gehört zu einer Seite und kann in Ausdrücken im Petri-Netz verwendet werden. Sie wurden nicht in der Definition der speziellen Petri-Netze angegeben, da sie nur zur Vereinfachung von Ausdrücken dienen und durch rein textuelle Ersetzung aufgelöst werden können. Ihre grafische Repräsentation besteht aus dem Namen, einem Gleichheitszeichen und dem Ausdruck.

Objektbezeichnungen und ihre Sichtbarkeit

Die Objekte auf einer Seite eines speziellen Petri-Netzes müssen zu verschiedenen Zwecken eindeutig referenzierbar sein. Dazu ist es notwendig, daß jedes Objekt einen auf der Seite eindeutigen Namen besitzt. Zu den Objekten gehören Stellen, Transitionen und Definitionen. Die Eindeutigkeit des Namens wird durch die eingesetzte Benutzungsoberfläche erzwungen. Eine Ausnahme sind hier die Modelle, die Arbeitspläne beschreiben. In ihnen können dieselben Objekte mehrfach auftreten, da z.B. dieselbe Maschine mehrere Operationen an einem Werkstück nacheinander ausführen muß. In diesem Fall treten mehrere Transitionen auf, die dieselbe Maschine bezeichnen und demnach auch identische Namen tragen. Dann ist es gestattet, gleiche Namen zu verwenden, da ja die Objekte gleichen Namens in der Realität tatsächlich den selben Betriebsmitteln entsprechen. Es handelt sich hier also nur um verschiedene Verweise auf dasselbe Objekt.

Um Objekte eindeutig bezeichnen zu können, die Objektamen aber nur auf ihrer Seite eindeutig sind, wird eine Notation ähnlich der in Dateisystemen eingeführt. Die Primärseite wird dabei mit **** bezeichnet. Da jede Unterseite einer Vertretertransition auf der Oberseite zugeordnet ist, können Seiten über den „Pfad“ der Vertretertransitionen eindeutig bezeichnet werden. Eine Seite, die einer Vertretertransition **Maschine1** untergeordnet ist, die sich wiederum auf der Unterseite einer Transition **FertigungszelleA** auf der Primärseite befindet, wird also mit **\FertigungszelleA\Maschine1** bezeichnet. Eine Stelle **Bearbeitungsplatz** auf dieser Seite heißt entsprechend **\FertigungszelleA\Maschine1\Bearbeitungsplatz**.

Für die Verwendung von Objekten ohne Pfadangabe gibt es außerdem folgende **Sichtbarkeitsregeln**:

- Normalerweise ist mit einem Objekt ohne Pfadangabe das auf derselben Seite gemeint, in dem der Objektname verwendet wird.
- Ähnlich wie in modularen Programmiersprachen können Objekte ohne Pfadangabe referenziert werden, die sich in einem der umgebenden Blöcke (hier: in einer der Oberseiten) befinden. Gibt es mehrere solcher Objekte, wird dasjenige gewählt, das sich auf der am tiefsten in der Seitenhierarchie befindlichen Unterseite befindet. Lokale Definitionen überschreiben demnach globalere.
- Objekte auf einer die Struktur modellierenden Seite sind außerdem auf den dazugehörigen Arbeitsplanseiten bekannt, als ob sie dort definiert wurden. Beide Modellteile beschreiben ja dieselben realen Objekte, die Arbeitsplaninformationen können darum hier Standardwerte aus dem Strukturmodell überschreiben, nicht aber umgekehrt.
- Definitionen, die auf einer assoziierten Arbeitsplanseite spezifiziert wurden, sind in keiner Strukturseite sicht- oder verwendbar.

3.2 Getrennte Modellierung von Fertigungssystem und Arbeitsplänen

Eine zentrale Anforderung an die Modellierung eines Fertigungssystems ist die Trennung zwischen Ressourcen- und Arbeitsplanbeschreibung im Modell. Dies dient nicht nur der besseren Verständlichkeit, sondern ist vorteilhaft, weil Arbeitspläne sehr viel häufiger geändert werden als z.B. das Layout eines Fertigungssystems. Eine grundlegende Eigenschaft der in dieser Arbeit vorgestellten Modellierungsmethode ist darum die getrennte Modellierung beider Aspekte mit spezialisierten Petri-Netzen. In diesem Kapitel wird anhand des im Abschnitt 2.2 dargestellten Beispiels die Vorgehensweise gezeigt. Dabei werden auch die Eigenheiten der verwendeten Netzarten im Vergleich zu der in 3.1.1 definierten Netzklasse erläutert.

Die Beschreibung eines Fertigungssystems setzt sich aus je einem speziellen Petri-Netz der Struktur und denen der Arbeitspläne zusammen. Jedes der Modelle besitzt eine Primärseite, von der aus durch Vertretertransitionen eine hierarchische Verfeinerung möglich ist. Das Strukturmodell beschreibt die strukturellen, arbeitsplanunabhängigen Eigenschaften des zu modellierenden Fertigungssystems. Die Modelle der Arbeitspläne beschreiben die zur Fertigung eines Produktes notwendigen Arbeitsschritte mit einem Pfad durch die Struktur. Sie können darum nur Objekte aus der Struktur enthalten. Die speziellen Petri-Netze der Arbeitspläne werden als mit dem Strukturmodell assoziierte Petri-Netze bezeichnet.

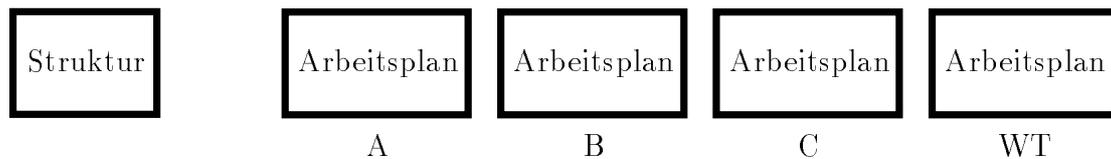


Abbildung 3.1: Verwaltung von Struktur- und Arbeitsplanmodellen

Zuerst wird die Struktur eines Fertigungssystems modelliert. Die dafür verwendete Netzart unterscheidet sich in den folgenden Punkten von der im Abschnitt 3.1.1 gegebenen Definition:

- Den Transitionen ist keine Übergangstabelle zugeordnet, da die verschiedenen Schaltmöglichkeiten zu diesem Zeitpunkt noch nicht bekannt sind.
- Die Transitionen besitzen dafür eine Standardschaltzeitverteilung und ein Standardgewicht. Diese ergeben sich aus den Eigenschaften des modellierten Betriebsmittels (Geschwindigkeit eines Transportmittels, normale Bearbeitungszeit einer Maschine). Die Werte werden bei der Generierung eines Gesamtmodells verwendet, wenn kein spezieller Wert im Arbeitsplanmodell angegeben wurde.

Die getrennte Modellierung von Struktur und Arbeitsplänen wird auch durch das entwickelte Softwarewerkzeug unterstützt, auf das im Kapitel 7 näher eingegangen wird. Die oberste Darstellungsebene des Beipielmodells aus Kapitel 2.2 wird im Bild 3.1 gezeigt. Das linke Kästchen steht für das Modell der Struktur des Fertigungssystems, von der es immer nur genau ein Modell gibt. Rechts davon ist für jedes Werkstück, das im Fertigungssystem auftritt, ein eigenes Arbeitsplanmodell vorhanden; im Beispiel also für die Teile **A**, **B**, **C** und die Werkstückträger **WT**. Durch einen Doppelklick auf eins der Kästchen gelangt man zum darunterliegenden Modell.

Strukturmodell des Beispiel-Fertigungssystems

Das Bild 3.2 zeigt die Primärseite des Modells des Fertigungssystem-Beispiels aus Kapitel 2.2. Entsprechend dem Prinzip der vorgeschlagenen Methode orientiert sich das Modell soweit wie möglich an der realen Struktur des Fertigungssystems. Das Modell kann daher z.B. aus einer Skizze abgeleitet werden, wie sie für das Beispiel in Bild 2.1 auf Seite 11 dargestellt ist. Durch die Ähnlichkeit der Bilder und die Benennung der Modellobjekte ist die Zuordnung der Betriebsmittel zu den Objekten im Modell möglich. Dabei entsprechen im einfachsten Fall Maschinen bzw. Transporteinrichtungen Transitionen und Lager Stellen.

Wie im Bild zu sehen, werden einfache Stationen durch normale Transitionen modelliert (im Beispiel **Roboter** und **Handarbeit**). Kompliziertere Betriebsmittel sind als Vertretertransition dargestellt, denn ihr Verhalten läßt sich nicht allein durch das Schalten einer Transition modellieren. Auf der der Vertretertransition zugeordneten Unterseite kann das Betriebsmittel dann genauer beschrieben werden. Auf die Unterseiten wird später eingegangen. Die drei auftretenden Lager **LagerAB**, **LagerC** und **FertigLager** werden als Stellen mit den entsprechenden Kapazitäten abgebildet. Um nicht nur einen kurzzeitigen Betrieb bis

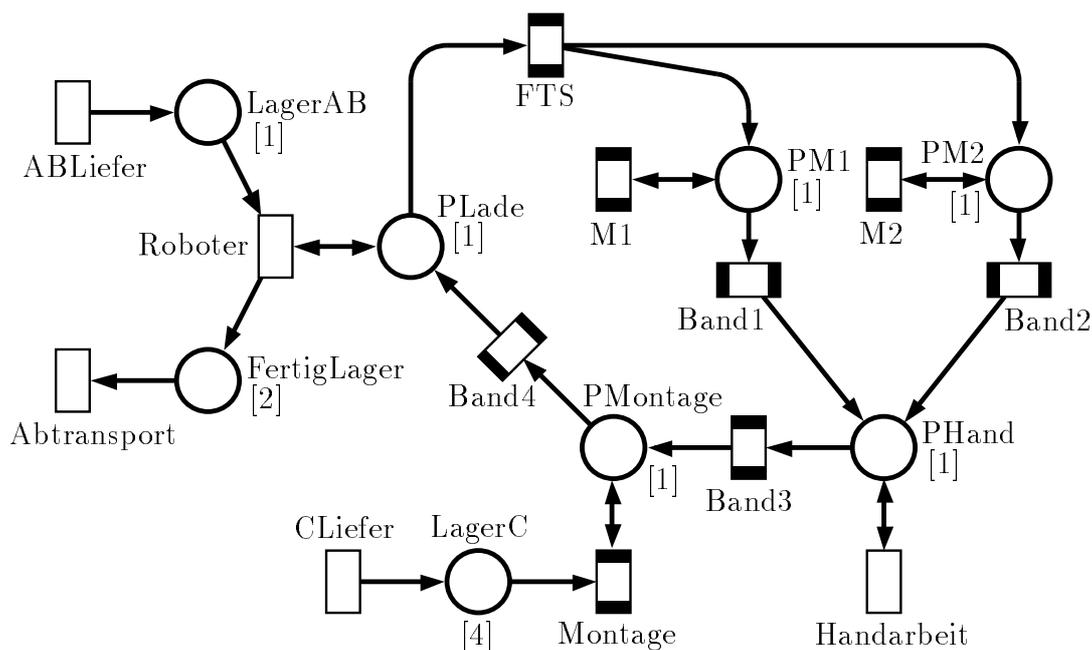


Abbildung 3.2: Strukturmodell des Beispiel-Fertigungssystems

zur Leerung der Eingangslager untersuchen zu können, muß auch die Anlieferung von Vorprodukten und der Abtransport der Fertigteile modelliert werden. Dies geschieht mit den Transitionen *ABLiefer*, *CLiefer* und *Abtransport*.

Stellen werden im Modell aber auch da benötigt, wo auf den ersten Blick im Fertigungssystem kein Lager vorhanden ist, und zwar für alle Orte, an denen sich ein Werkstück befinden kann. Bei jeder Maschine und jeder Bearbeitungsstation befindet sich im Normalfall ein Platz, auf der sich das gerade bearbeitete Werkstück befindet. Daraus ergeben sich die Stellen *PLade*, *PM1*, *PM2*, *PHand* und *PMontage*. Diese Stellen bezeichnen die Orte, an denen sich Werkstückträger bei einer Station im Fertigungssystem befinden können; ihre in eckigen Klammern angegebene Kapazität kann den Angaben in Abbildung 2.2 entnommen werden.

Da alle Stellen auf der Primärseite in unserem Fall potentielle Orte von Werkstücken modellieren, sind sie Produktstellen und werden dick gezeichnet. Das gleiche gilt demzufolge für die Kanten. Jede Kante steht entweder für einen möglichen Weg eines Produktes im Fertigungssystem in Pfeilrichtung (z.B. die Kante zwischen *PLade* und *FTS*) oder für die Möglichkeit, den Zustand einer Marke in einer Stelle als Ergebnis einer Bearbeitung zu verändern (z.B. die Kanten zwischen *M1* und *PM1*). Kanten des zweiten Typs treten typischerweise doppelt auf, also als Verbindung in beiden Richtungen zwischen Transition und Stelle. Das liegt daran, daß die Veränderung des Zustands einer Marke nicht direkt modelliert wird, sondern durch das Abziehen der alten Marke und gleichzeitiges Erzeugen der neuen Marke nachgebildet wird. Durch die Darstellung beider Richtungspfeile übereinander entstehen Doppelpfeile, die aber in Wirklichkeit zwei eigenständige Kanten sind.

Mit dem Strukturmodell werden alle „Fähigkeiten“ des Fertigungssystems dargestellt. Betriebsmittel, die hier nicht beschrieben wurden, sind im Modell nicht bekannt und können später nicht in den Arbeitsplänen verwendet werden. Die Attribute von Objekten im Strukturmodell beschreiben die unabhängig von den Arbeitsplänen geltenden Standardwerte, z.B.

für Bearbeitungszeiten. In unserem Beispiel müssen auf der Primärseite die Schaltzeitverteilungen und die Schaltbedingungen für die einfachen Transitionen angegeben werden. Die Schaltzeitverteilungen enthält die folgende Tabelle; Schaltbedingungen sind hier nicht notwendig und müssen darum auch nicht spezifiziert werden. Sie werden dann immer als Wahr angenommen.

<i>Transition</i>	<i>modelliert</i>	<i>Schaltzeitverteilung</i>
ABLiefer	Anlieferung der Teile A und B	Exponential(30)
CLiefer	Anlieferung von C-Teilen	Exponential(20)
Roboter	Roboter an der Beladestation	Exponential(10)
Abtransport	Abtransport fertiger Teile	Exponential(30)
Handarbeit	manueller Arbeitsgang	Exponential(60)

Die Syntax zur Angabe von Schaltzeitverteilungen ist im Anhang D enthalten. Die Angabe von **Exponential(30)** bedeutet eine exponentiell verteilte Schaltzeit, deren mittlere Schaltzeit 30 beträgt. Die Schaltzeiten seien in Sekunden angegeben. Auf der Primärseite könnte auch gleich der Anfangszustand durch die Initialisierungsfunktion der Stellen angegeben werden. Darauf wird hier verzichtet.

Nachdem zuerst die Primärseite des Strukturmodells modelliert wurde, muß die Struktur auf tieferen Ebenen der Hierarchie genauer beschrieben werden. Dazu wird als nächstes das Modell des fahrerlosen Transportsystems verfeinert, um zu zeigen, wie das manuell geschieht. Für die anderen Vertretertransitionen, die die Maschinen, den Montageplatz und die Förderbänder modellieren, werden im folgenden Abschnitt 3.3 beispielhaft generische Teilmodelle aus einer Bibliothek verwendet. Die Beschreibung des Aufbaus der entsprechenden Unterseiten wird darum erst dort angegeben.

Die Vertretertransition **FTS** auf der Primärseite entspricht dem fahrerlosen Transportsystem, das Werkstückträger von der Ladestation (Stelle **PLade**) abholt und dann entweder zur Bohrmaschine (Stelle **PM1**, Transition **M1**) oder zur Fräse (Stelle **PM2**, Transition **M2**) befördert. Auf der in Bild 3.3 dargestellten Unterseite befinden sich darum die gestrichelt gezeichneten Anschlußstellen **PLade**, **PM1** und **PM2**, die die Verbindung zu den die Vertretertransition **FTS** umgebenden Stellen auf der Primärseite herstellen. Die Anschlußstellen wurden hier der Einfachheit halber genauso benannt wie die entsprechenden Sockelstellen. Die Wahl der Namen ist beliebig, die Zuordnung der Anschluß- zu den Sockelstellen erfolgt unabhängig davon. In der Definition ist die Zuordnung über die Anschlußzuweisungsfunktion **AZ** gegeben. Da die Anschlußstellen nur Verweise auf die Sockelstellen sind, können auf der Unterseite die Attribute bis auf den Namen nicht unabhängig von den Sockelstellen geändert werden.

Ganz oben auf der Seite befindet sich die folgende Definition: **Leer = (#AufFTS = 0)**. Es handelt sich hier um einen markierungsabhängigen booleschen Ausdruck, der angibt, ob das **FTS**-Fahrzeug leer ist oder gerade einen Werkstückträger transportiert. Davon ist die Definition **Voll = !Leer** abgeleitet; **!** steht für die logische Negierung. Die Ausdrücke **Leer** und **Voll** können dann innerhalb des Modells verwendet werden.

Der obere Teil des Modells enthält Produktstellen und modelliert den Aufnahmeplatz auf dem **FTS**-Fahrzeug (Stelle **AufFTS** mit der Kapazität Eins) sowie das Be- und Entladen an den verschiedenen anfahrbaren Orten im Fertigungssystem. Letzteres wird durch die Transitionen **Beladen**, **EntladenM1** und **EntladenM2** abgebildet.

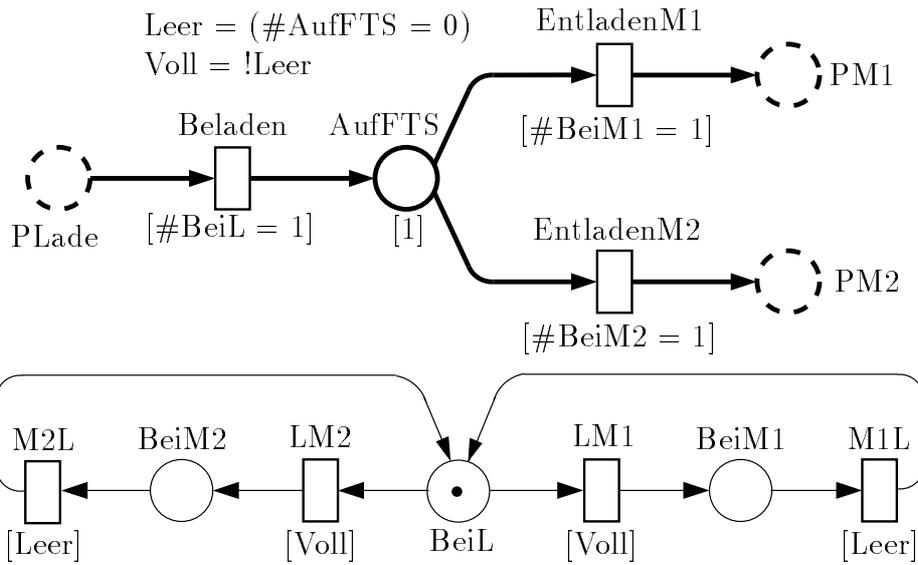


Abbildung 3.3: Unterseite des fahrerlosen Transportsystems

Im unteren Modellteil wird mit Elementarstellen der aktuelle Ort (Stellen **BeiL**, **BeiM1** und **BeiM2**) und die Bewegung (Transitionen **LM2**, **M2L**, **LM1** und **M1L**) des FTS-Fahrzeugs modelliert. Die Namen der Transitionen bestehen aus dem Anfangs- und Zielort der Bewegung, wobei ein „L“ für die Ladestation steht.

Das Zusammenspiel beider Modellteile spezifiziert die Funktionsweise: Steht das Fahrzeug leer an der Ladestation und befindet sich ein Werkstückträger in **PLade**, kann Transition **Beladen** schalten und „lädt“ damit den Werkstückträger auf das Fahrzeug (Stelle **AufFTS**). Durch die Kapazität Eins ist gesichert, daß sich nur ein Teil auf dem Fahrzeug befindet. Nun kann eine der Transitionen **LM1** oder **LM2** schalten. Welche gewählt wird, hängt von der Art des Werkstücks ab und wird erst später im entsprechenden Arbeitsplanmodell beschrieben. Durch die Schaltbedingung **[Voll]** wird gesichert, daß die Transitionen erst schalten, wenn das Fahrzeug beladen ist. Wenn das Fahrzeug dann beispielsweise durch das Schalten von **LM2** am Ziel **M2** angekommen ist, befindet sich die farblose Marke in **BeiM2**. Damit wird **EntladenM2** schaltfähig, denn die Transition besitzt die Schaltbedingung **[#BeiM2 = 1]**. Nachdem **EntladenM2** geschaltet hat, ist der transportierte Werkstückträger in Stelle **PM2**, der Ausdruck **Leer** wird wahr und damit Transition **M2L** schaltfähig. Das Schalten von **M2L** modelliert dann das Zurückfahren des Fahrzeugs zur Ladestation. Weitere Informationen zu den Modellelementen enthält die nachfolgende Tabelle.

<i>Transition</i>	<i>modelliert</i>	<i>Schaltzeitverteilung</i>
Beladen	Beladen des Fahrzeugs	Exponential(2)
EntladenM1	Entladen des Fahrzeugs bei M1	Exponential(2)
EntladenM2	Entladen des Fahrzeugs bei M2	Exponential(2)
LM1	Fahrt Ladestation→M1	Exponential(10)
M1L	Fahrt M1→Ladestation	Exponential(10)
LM2	Fahrt Ladestation→M2	Exponential(12)
M2L	Fahrt M2→Ladestation	Exponential(12)

Arbeitsplanmodelle des Beispiel-Fertigungssystems

Nun werden die Arbeitspläne mit den speziellen Petri-Netzen modelliert. Dabei wird wie bei der Struktur mit den Primärseiten begonnen und dann die Verfeinerung der Vertretertransitionen vorgenommen. Die Modelle der Arbeitspläne beschreiben den bei der Fertigung eines Produktes zu durchlaufenden Pfad durch das Strukturmodell. Sie werden aufgrund ihrer engen Beziehung zum Strukturmodell als zu diesem **assoziierte Petri-Netze** bezeichnet.

Daraus ergeben sich folgende Besonderheiten und Regeln für die Netzart zur Beschreibung der Arbeitspläne im Unterschied zur in Abschnitt 3.1.1 gegebenen Definition:

- Für jeden Arbeitsplan eines Fertigungssystems muß ein eigenes Arbeitsplanmodell erstellt werden. Jedes dieser Modelle beschreibt ausschließlich Fertigungsschritte, die zur Herstellung eines Produktes notwendig sind. Der Arbeitsplan eines Produktes darf aber auf mehrere Modelle aufgeteilt werden, wenn dies z.B. für eine übersichtlichere Darstellung notwendig ist.
- Ein assoziiertes Petri-Netz enthält nur Stellen, Transitionen und Kanten, zu denen es eine Entsprechung im Strukturmodell gibt. Diese Zuordnung beginnt mit den Primärseiten und setzt sich auf allen untergeordneten Seiten fort. Der absolute Pfad eines Objektes in einem assoziierten Petri-Netz entspricht dem Pfad des zugeordneten Objektes im Strukturmodell. Damit ist gesichert, daß nur in der Struktur des Fertigungssystems tatsächlich vorhandene Ressourcen in einem Arbeitsplan verwendet werden. Durch die Verwendung eindeutiger Namen auf jeder Seite eines speziellen Petri-Netzes ist die Zuordnung der Arbeitsplan- zu den Strukturobjekten eindeutig, denn es müssen dieselben Namen wie im Strukturmodell verwendet werden. Wie im Abschnitt 3.1.2 beschrieben, sind für die assoziierten Petri-Netze gleiche Namen auf einer Seite zugelassen, wenn sie dasselbe strukturelle Objekt bezeichnen. Kanten sind über die mit ihnen verbundenen Objekte referenzierbar.
- Den Transitionen ist keine Übergangstabelle zugeordnet, da die verschiedenen Schaltmöglichkeiten zu diesem Zeitpunkt noch nicht bekannt sind. Sie besitzen dafür eine Schaltbedingung, eine Schaltzeitverteilung und ein Gewicht. Diese ergeben sich aus den speziellen Eigenschaften des modellierten Betriebsmittels (z.B. Bearbeitungszeit einer Maschine) bei der Ausführung des modellierten Fertigungsschritts. Die Werte sind also vom Arbeitsplan abhängig und werden bei der Generierung des Gesamtmodells verwendet. Sie überschreiben dabei die Standardwerte aus dem Strukturmodell. Für zeitbehaftete Transitionen mit der Schaltsemantik *exclusive server* kann außerdem die Auswahlwahrscheinlichkeit des Übergangs angegeben werden.
- Da die assoziierten Modelle nur zusätzliche oder spezielle Angaben spezifizieren, dürfen keine strukturellen Objektattribute aus dem zugeordneten Strukturmodell verändert werden. Die Angabe von Initialisierungsfunktion, Kapazität oder Schaltsemantik ist demnach nicht zugelassen. Weil sich der Arbeitsplan auf die Werkstücke bezieht, dürfen keine Elementarstellen und -kanten in assoziierten Modellen verwendet werden. Die elementaren Stellen und Kanten werden nur für die Beschreibung von strukturellen Eigenschaften verwendet.

- Da jedes assoziierte Petri-Netz einen Pfad durch die Struktur beschreibt, darf das Modell nur aus einer linearen Abfolge von Transitionen und Stellen bestehen, damit die Reihenfolge der Fertigungsschritte eindeutig festgelegt ist. Davon ausgenommen sind lediglich alternative Fertigungsabläufe sowie Montage- und Demontagevorgänge, deren Beschreibung weiter unten am Beispiel gezeigt wird. Der Beginn und das Ende eines assoziierten Petri-Netzes sollte entweder in einem Ein- oder Ausgang des Systems liegen, oder aber eine Schnittstelle zu einem anderen Arbeitsplanmodell darstellen.
- An den ausschließlich auftretenden Produktkanten ist der Typ bzw. die Farbe der transportierten oder bearbeiteten Marken verzeichnet. Um auch eine von Eins abweichende Menge von verarbeiteten Teilen angeben zu können, sind dafür allgemein Multimengen von Produktmarken zugelassen. Im Normalfall treten als Bezeichnungen an den Kanten eines Arbeitsplanmodells nur Farben auf, die dasselbe Produkt in unterschiedlichen Bearbeitungsstufen beschreiben. Montage- und Demontagevorgänge stellen eine Ausnahme dar, denn sie betreffen verschiedene Produkte. Vorgänge dieser Art sollten im Arbeitsplan desjenigen Produktes modelliert werden, an das ein anderes Werkstück anmontiert oder von dem eines abgebaut wird, so daß danach das ursprüngliche Produkt in einem anderen Bearbeitungszustand vorliegt. Dies ist sinnvoll, da z.B. im Arbeitsplan eines Hilfsstoffes nicht unbedingt bekannt ist, wo er nach seiner Fertigstellung benötigt und verbraucht wird. Nur am Beginn eines Arbeitsplans, bei Montage- und Demontagevorgängen dürfen fremde Produktbezeichnungen in einem Arbeitsplanmodell auftreten.
- Die Produktbeschreibungen an allen Kanten, die von gleichnamigen Stellen eines Arbeitsplanmodells ausgehen, müssen unterschiedlich sein. Ansonsten wäre der Arbeitsplan mehrdeutig. Jede Transition muß demnach entweder den Ort (Transport), die Variante (Entscheidung über eine Fertigungsalternative) oder den Zustand (Bearbeitung) einer Marke verändern. Außerdem müssen alle Produktbeschreibungen, die an den Ein- und Ausgangskanten einer Stelle stehen, in Produkt und Bearbeitungsstufe identisch sein, da in einer Stelle das Werkstück nicht verändert werden kann. Unterschiedliche Varianten eines Werkstücks dürfen an den Kanten, die mit einer Stelle verbunden sind, nur am Beginn einer Variante auftreten; dort sind sie aber erforderlich. Die genannten Einschränkungen gelten nur für das Produkt, zu dem der Arbeitsplan gehört. Bei Montagevorgängen verbrauchte und bei Demontage erzeugte Werkstücke bilden eine Ausnahme von diesen Einschränkungen.
- Fertigungsalternativen im Arbeitsplanmodell sind daran zu erkennen, daß von einer Stelle mehrere Kanten zu Transitionen hinführen. Jedes Werkstück kann nur über eine dieser Kanten transportiert oder bearbeitet werden, es ist also eine Auswahl zu treffen. Dafür werden zwei Konstrukte zur Verfügung gestellt:
 - Einer solchen Kante kann eine Bedingung, die in eckigen Klammern dargestellt wird, zugeordnet sein. Dieser boolesche markierungsabhängige Ausdruck wird für jede der Folgekanten einer Stelle ausgewertet. Ein Werkstück darf nur Wege weiterverfolgen, deren Bedingung zum Zeitpunkt der Entscheidung wahr ist. Wird keine Bedingung angegeben, wird sie als Wahr angenommen.

- Außerdem kann eine Wahrscheinlichkeit angegeben werden. Dieser reellwertige Ausdruck wird für alle Kanten, deren Bedingung wahr ist, ausgewertet. Das Ergebnis ist jeweils die relative Wahrscheinlichkeit dafür, daß dieser Pfad betreten wird. Die Wahrscheinlichkeit für das Wählen der Variante ergibt sich demnach aus dem Wahrscheinlichkeitsausdruck geteilt durch die Summe der Ausdrücke aller der Kanten, deren Bedingung wahr ist. Wenn keine Wahrscheinlichkeit angegeben ist, wird sie als Eins angenommen.

Um zwischen Werkstücken in verschiedenen Alternativen unterscheiden zu können, die ansonsten denselben Bearbeitungszustand haben, werden die Variantenbezeichnungen verwendet. Da bei Beginn einer Alternative Marken desselben Zustands eine Stelle über mehrere Wege verlassen, müssen sie durch verschiedene Variantennamen unterscheidbar sein. Der Beginn einer Alternative ist demnach auch daran zu erkennen, daß Kanten mit unterschiedlichen Variantenbezeichnungen mit derselben Stelle verbunden sind. Weitere Erläuterungen dazu enthält Abschnitt 3.4.

- Das Ende einer Fertigungsalternative ist dadurch gekennzeichnet, daß mehrere Kanten zu einer Stelle führen. Alle diese Kanten müssen den gleichen Markentyp besitzen; eine eventuell notwendige Änderung der Variante einer Marke muß also schon am Ende des vorherigen Arbeitsschritts erfolgen.
- Um keine versteckten, d.h. über das Modell verstreuten Konflikte im Arbeitsplan entstehen zu lassen, müssen alle Kanten mit gleichen Produktbezeichnungen, die zu gleichnamigen Stellen führen, tatsächlich mit derselben Stelle verbunden sein. Dies ist typischerweise am Ende von alternativen Fertigungsabläufen der Fall und kann durch das Zusammenlegen der problematischen Stellen erfüllt werden. Von dieser Einschränkung sind wiederum Montage- und Demontagevorgänge ausgenommen.

Im Bild 3.4 ist die Primärseite des Arbeitsplanmodells für das Werkstück A dargestellt. Es wurde aus dem in Bild 2.4 dargestellten Arbeitsplan von Teil A abgeleitet. Dabei muß das Strukturmodell (Bild 3.2) beachtet werden, um die gleichen Transitionen und Stellen zu verwenden. Die an den Kanten eingezeichneten Werkstückarten und Bearbeitungsschritte sind ebenfalls dem Arbeitsplan entnommen.

Entsprechend den beschriebenen Besonderheiten der assoziierten Netze wurde das Arbeitsplanmodell soweit wie möglich linear gestaltet. Dadurch muß z.B. bei einer Bearbeitung durch **M2** die verbundene Stelle **PM2** zweimal hintereinander dargestellt werden. Das Modell ähnelt dadurch stark der Beschreibung des Arbeitsplans in Bild 2.5.

Eine Ausnahme stellt dabei die Montage des Werkstücks **A.Geliefert** mit dem leeren Werkstückträger **WT.leer** durch den Roboter dar. Damit wird nachgebildet, daß ein Werkstück nur gemeinsam mit einem Träger durch das Fertigungssystem transportiert wird. Am Ende des Arbeitsplans wird der Werkstückträger dann wieder durch den Roboter „demontiert“, und es entsteht das Teil **A.Fertig**.

Da beim Erstellen eines Arbeitsplanmodells ein Pfad durch das Strukturmodell beschrieben wird, kann ein Softwarewerkzeug eine Unterstützung folgender Art anbieten: Wenn der Modellierer den Arbeitsplan bei der Transition **ABLiefer** beginnt, ist danach in der Struktur nur der Weg zur Stelle **LagerAB** möglich. Diese Stelle und die entsprechende Kante können also

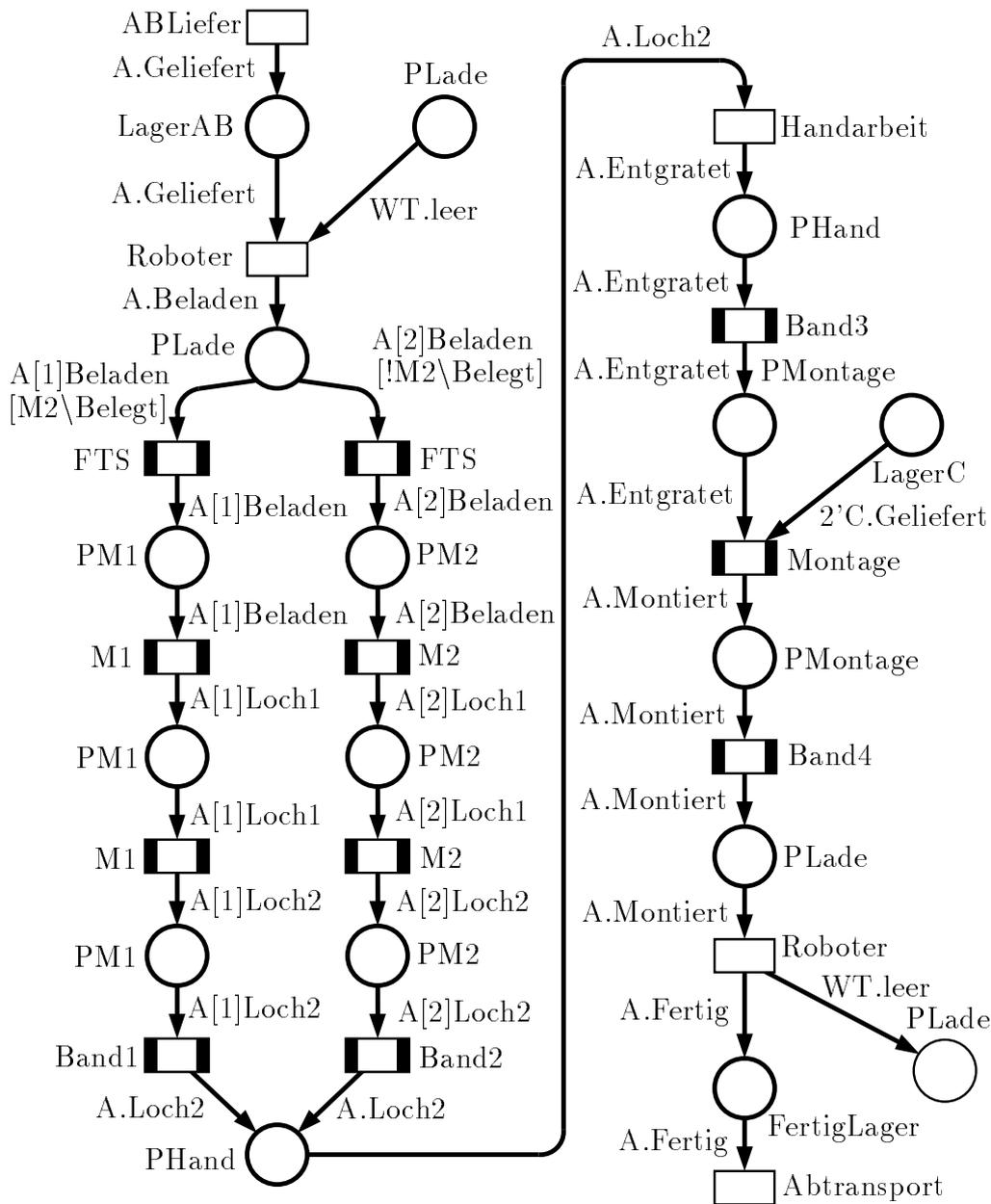


Abbildung 3.4: Arbeitsplan A für das Beispiel-Fertigungssystem

vom Werkzeug automatisch erzeugt werden. In Fällen, in denen es mehrere Möglichkeiten gibt, kann eine Auswahl aus den durch die Struktur zulässigen Folgestellen oder -transitionen angeboten werden, aus der der Modellierer nur auszuwählen braucht. Der Markentyp an den Kanten kann als der vorherige angenommen werden, so daß der Modellierer nur bei einer Änderung des Bearbeitungszustands etwas eingeben muß. Montage- und Demontagevorgänge müssen gesondert behandelt werden.

Durch die Montagestation werden in einem Arbeitsschritt auf der rechten Seite des Modells zwei Teile des Typs C aus LagerC entnommen und mit A.Entgratet montiert, wobei A.Montiert entsteht. Durch die Angabe von C.Geliefert in diesem Arbeitsplan, der das Werkstück A beschreibt, ist eine Montage zu erkennen. Wie die C-Teile in die Stelle LagerC

gelangen, darf nicht hier, sondern muß im Arbeitsplan von C beschrieben werden. Der Arbeitsplan beginnt und endet, wie gefordert, in einem Systemein- bzw. Ausgang (**ABLiefer** bzw. **Abtransport**).

Bei der Herstellung der zwei Aufnahmelöcher gibt es alternativ die Möglichkeiten, die Bohrmaschine oder die Fräse einzusetzen. Diese Alternative wird im Modell genauso wie im Arbeitsplan beschrieben und ist daran zu erkennen, daß das Werkstück in derselben Bearbeitungsstufe auf zwei verschiedenen Wegen die Stelle **PLade** verlassen kann. Dabei wird dem Werkstück zur Unterscheidung eine der Versionsnummern 1 oder 2 gegeben. Da die Werkstücke nach Durchlaufen der Alternative wieder denselben Fertigungsschritten unterliegen, enden beide Alternativen in der Stelle **PHand**, vor der die Versionsnummern der Werkstücke auf 0 zurückgesetzt werden.

Wenn sich ein Werkstück vom Typ **A.Beladen** in der Stelle **PLade** befindet, muß entschieden werden, welche Alternative (Maschine 1 oder 2) für das Teil ausgewählt wird. Dies kann mit Hilfe von Bedingungen geschehen, die die möglichen Wege einschränken. In unserem Beispiel sollen die Werkstücke **A.Beladen** vorrangig auf Maschine 2 bearbeitet werden, da Maschine 1 außerdem noch für die Bearbeitung der Teile vom Typ B verantwortlich ist. Maschine 1 soll demnach nur dann ausgewählt werden, wenn Maschine 2 bereits belegt ist. Dies wird mit der Bedingung `[M2\Belegt]` an der zu **M1** führenden Kante angegeben. Über den Pfad `M2\Belegt` wird das Objekt **Belegt** auf der zur Vertretertransition **M2** gehörenden Seite angesprochen. **Belegt** ist dort eine Definition und steht für einen markierungsabhängigen booleschen Ausdruck, der wahr ist, wenn die Maschine gerade mit einem Werkstückträger belegt ist. Da diese Unterseite ein Bibliotheksmodul ist, wird sie weiter unten im Abschnitt 3.3 näher erläutert. An der zu **M2** führenden Kante steht die Negation desselben Ausdrucks, so daß immer genau eine von beiden Kanten gewählt wird. Dies ist typischerweise dann der Fall, wenn eindeutige Entscheidungen über die Auswahl einer Alternative aufgrund der aktuellen Situation im Fertigungssystem gefällt werden können.

Wie beschrieben, können hier für die einfachen Transitionen Attribute angegeben werden, die vom Arbeitsplan abhängen. Weggelassene Attribute werden später als Standardwert aus dem Strukturmodell übernommen. Im Beispiel soll von den alle 30 Sekunden angelieferten Teilen (Transition **ABLiefer**) 70 Prozent vom Typ A sein. Darum wird die Auswahlwahrscheinlichkeit des Übergangs, der der Anlieferung eines A-Teils entspricht, von Transition **ABLiefer** auf 0.7 gesetzt, da es sich hier um eine Transition mit der Schaltsemantik *exclusive server* handelt. Die mittlere Schaltzeit der Transition **Handarbeit** wird für den hier auszuführenden Arbeitsgang des Entgratens auf 40 Sekunden gesetzt.

Bild 3.5 zeigt die Primärseite des Arbeitsplanmodells für das Werkstück B. Es wurde wie das Arbeitsplanmodell von Teil A aus dem in Bild 2.4 dargestellten Arbeitsplan abgeleitet. Die an den Kanten eingezeichneten Werkstückarten und Bearbeitungsschritte sind wiederum dem Arbeitsplan entnommen.

Die Erläuterungen zum oben beschriebenen Arbeitsplanmodell von Teil A gelten hier entsprechend. Der größte Unterschied zwischen beiden assoziierten Modellen liegt darin, daß für Teil B am Handarbeitsplatz das Ergebnis des Bohrens geprüft wird. Abhängig vom Prüfergebnis durchläuft das Werkstück eine von zwei möglichen Fertigungsabläufen. Wenn die Prüfung negativ ausfiel, muß das Werkstück nachgebohrt werden; wenn nicht, kann dieser Zweig übersprungen werden. In diesem Fall ist es nur am Trennungspunkt der beiden Ferti-

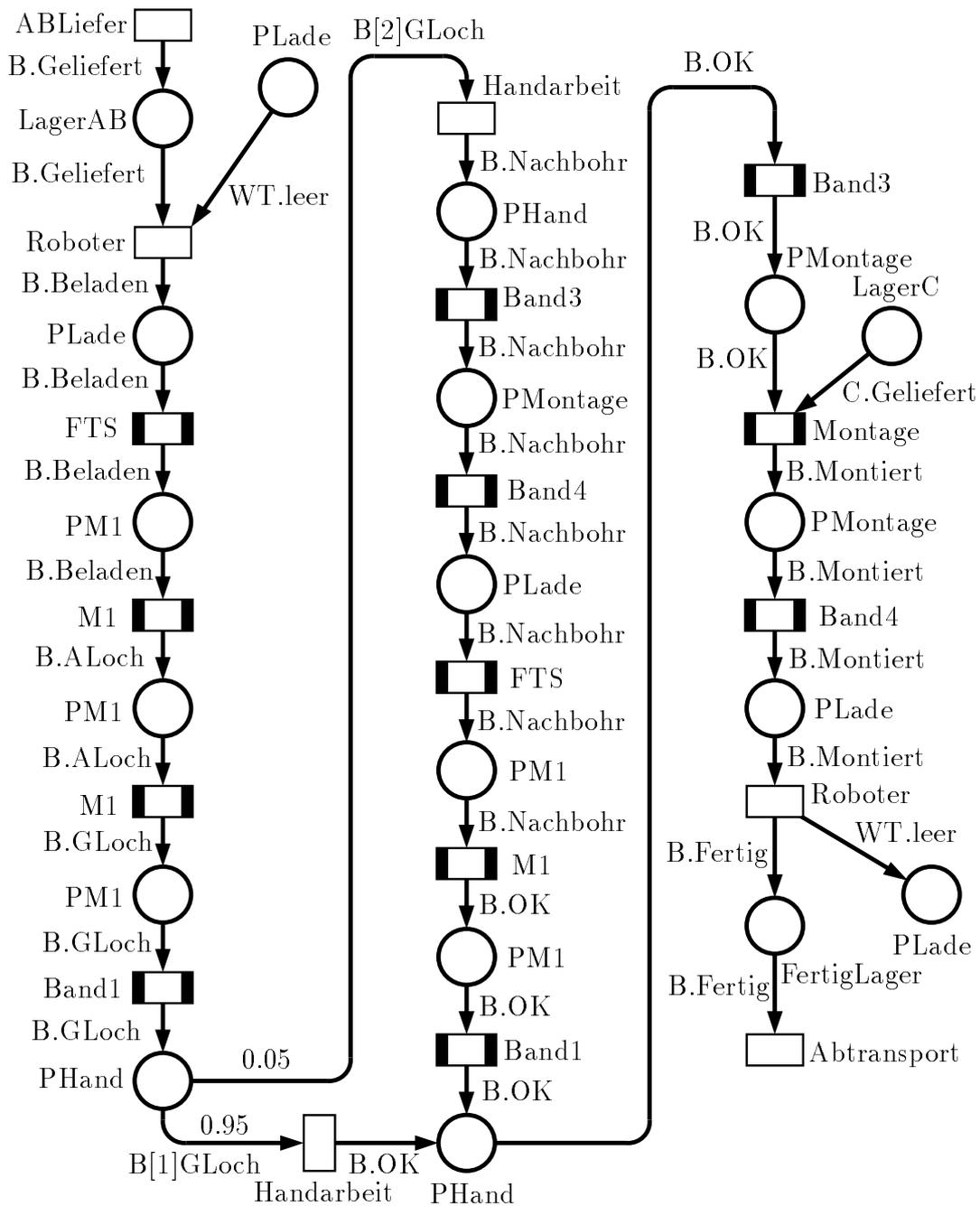


Abbildung 3.5: Arbeitsplan B für das Beispiel-Fertigungssystem

gungsalternativen (bei Stelle PHand) notwendig, den Teilen vom Typ B.GLoch verschiedene Variantennummern zu geben, denn sie unterscheiden sich ja ansonsten bereits durch ihren Bearbeitungszustand eindeutig. In der Stelle PHand laufen beide Varianten wieder zusammen, denn das Werkstück hat hier wieder denselben Zustand. Nach den Regeln für Arbeitsplanmodelle müssen die Kanten mit dem Markentyp B.OK, die beide zur Stelle PHand führen, auch zur selben Stelle gehen und nicht zu zwei verschiedenen Stellen mit dem Namen PHand. Damit wird erzwungen, daß am Ende der Fertigungsalternative der Arbeitsplan auch wieder zusammengeführt wird.

Auf den ersten Blick ist es vielleicht überraschend, daß sich die beiden Varianten bereits vor der Transition **Handarbeit** trennen, durch die ja erst die Entscheidung gefällt wird. Das liegt daran, daß jede Transition genau einen Vorgang modelliert. Der Vorgang des Prüfens mit negativem Ausgang und der mit positivem Ausgang sind zwei verschiedene Vorgänge. Nur einer kann ausgeführt werden, für jedes Teil vom Typ **B.GLoch** in der Stelle **PHand** gibt es daher einen Konflikt zwischen beiden möglichen Vorgängen. Im Arbeitsplanmodell A wurde gezeigt, wie alternative Routen abhängig von Bedingungen ausgewählt werden können. Dies ist hier nicht möglich, denn der Ausgang der Prüfung hängt nicht vom Zustand des Fertigungssystems ab. Es werden also keine Bedingungen angegeben, statt dessen ist eine zufällige Auswahl mit gewissen Wahrscheinlichkeiten notwendig. Diese werden an den Kanten angegeben und spezifizieren die relative Wahrscheinlichkeit, mit der der jeweilige nächste Fertigungsschritt ausgeführt wird. Wenn also in unserem Beispiel fünf Prozent aller Teile **B.GLoch** nachgebohrt werden müssen, stehen die Werte 0.05 und 0.95 an den Kanten. Im Abschnitt 3.4 werden diese Wahrscheinlichkeiten in das Gesamtmodell eingesetzt, so daß das beabsichtigte Verhalten erreicht wird.

Außerdem werden für die einfachen Transitionen Attribute angegeben, die vom Arbeitsplan abhängen und sich vom Standardwert aus dem Strukturmodell unterscheiden. Die Auswahlwahrscheinlichkeit des Übergangs der Transition **ABLiefer** muß hier auf 0.3 festgesetzt werden, denn nur 30 Prozent der angelieferten Teile sind vom Typ B. Die Dauer des Prüfvorgangs am Handarbeitsplatz soll jeweils im Mittel 20 Sekunden betragen und exponentialverteilt sein.



Abbildung 3.6: Arbeitsplan C für das Beispiel-Fertigungssystem

Bild 3.6 zeigt die Primärseite des Arbeitsplanmodells für Teile vom Typ C. Er ist sehr kurz und einfach, da nur die Anlieferung der C-Teile ins **LagerC** beschrieben wird. Die Schaltzeitverteilung der Transition **CLiefer** ist mit **Exponential(30)** angegeben, so daß alle 30 Sekunden zwei Teile des Typs C angeliefert werden.

Die Verwendung der C-Teile wurde ja bereits in den entsprechenden Arbeitsplänen A und B dargestellt. Da diese Seite keine Vertretertransitionen enthält, besitzt sie keine Unterseiten und stellt somit das vollständige assoziierte Modell für diesen Arbeitsplan dar.

Wie beschrieben werden die Produkte im Fertigungssystem auf Werkstückträgern transportiert, die im Arbeitsplanmodell mit den Werkstücken montiert werden. Häufig wird es aber auch leere Werkstückträger im System geben. Diese müssen weitertransportiert werden, um nicht den Weg für Werkstücke auf beladenen Trägern zu versperren. Diese Transportaufgaben sind Vorgänge im Fertigungssystem und müssen demnach auch mit einem eigenen Arbeitsplan modelliert werden. Da dieser Plan keine Bearbeitungsvorgänge enthält, ist er einfach aus dem geplanten Weg durch das Fertigungssystem ableitbar. Dabei wurde im Beispiel entschieden, die Werkstückträger der Einfachheit halber nur an Maschine 2 vorbei zu transportieren. Es wäre aber auch ohne weiteres möglich, hier eine Alternative wie im Arbeitsplan A zu verwenden. Es entsteht ein einfacher Rundkurs der Werkstückträger, dessen Arbeitsplan in Bild 3.7 dargestellt ist. Um den Plan linear zu halten, wurde der Kreis an

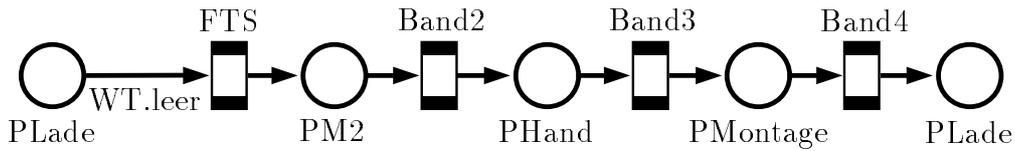


Abbildung 3.7: Arbeitsplan WT für das Beispiel-Fertigungssystem

einer (beliebigen) Stelle aufgespaltet. Da sich der Zustand der Marken **WT.leer** nicht ändert, wurde auf die Darstellung des Markentyps an den Kanten außer der ersten im Bild verzichtet. Alle Transitionen in diesem Modell sind Vertretertransitionen, es gibt auf dieser Ebene demnach nichts weiter zu tun. Genauere Angaben zum Verhalten können auf den Unterseiten gemacht werden.

Zu allen Vertretertransitionen in den assoziierten Arbeitsplanmodellen muß es entsprechende Unterseiten geben. Für Teilm Modelle aus einer Bibliothek sind diese zusammen mit der entsprechenden Strukturseite spezifiziert. Am Beispiel der Modelle der Maschinen, der Förderbänder und der Montagestation wird dies im folgenden Abschnitt 3.3 vorgeführt. Hier ist darum nur die Angabe der Unterseiten notwendig, die zum Strukturmodell des fahrerlosen Transportsystems gehören. Davon sind hier beispielhaft nur zwei dargestellt, die fehlenden Unterseiten modellieren den Transport der Teile **A[2]Beladen**, **B.Beladen**, **B.Nachbohr** und **WT.leer**. Sie gleichen den hier dargestellten bis auf den Typ des transportierten Werkstücks. Man kann sich die für die Modellierung notwendige Arbeit natürlich durch Kopieren und Ändern der Modelle erleichtern, trotzdem zeigt die Anzahl der notwendigen Modelle die Notwendigkeit von Bibliotheksmodulen, die instanziiert und in das Modell eingesetzt werden (siehe Abschnitt 3.3).

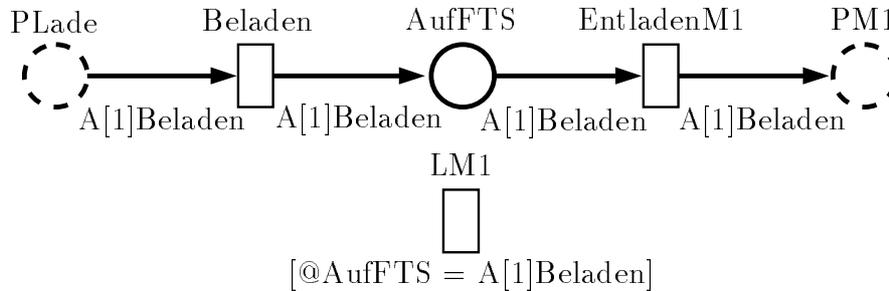


Abbildung 3.8: Arbeitsplan A Submodell fahrerloses Transportsystem

Bild 3.8 zeigt die Unterseite der Vertretertransition **FTS**, die **A[1]Beladen**-Marken von **PLade** nach **PM1** transportiert (siehe Primärseite des Arbeitsplanmodells **A** auf Seite 46). Diese Seite ist demnach assoziiert zum Strukturmodell des fahrerlosen Transportsystems in Bild 3.3. Hier wird beschrieben, wie der Transport eines speziellen Werkstücks im einzelnen vor sich geht. Im oberen Modellteil steht der Weg des transportierten Werkstücks von der Ladestation **PLade** über das **FTS**-Fahrzeug **AufFTS** bis zur Maschine **PM1**. Da hier nur der benutzte Pfad aus der Struktur angegeben wird, sind **EntladenM2**, **PM2** und die entsprechenden Kanten nicht verwendet worden. Weil der Zustand des Werkstücks durch den Transport nicht verändert wird, steht an allen Kanten **A[1]Beladen**. Die Namen der Anschlußstellen **PLade** und **PM1** sind beliebig, wurden hier aber wieder entsprechend den Namen der Sockelstellen gewählt.

Darunter ist die Transition LM1 aus dem „farblosen“ Teil des Strukturmodells eingezeichnet. Dies geschieht, um die spezielle Schaltbedingung [$\text{@AufFTS} = \text{A}[1]\text{Beladen}$] angeben zu können, die im Zusammenspiel mit den Schaltbedingungen auf der Strukturseite die korrekte Funktionsweise sichert. Mit dieser Schaltbedingung wird erreicht, daß das FTS-Fahrzeug sich erst von der Ladestation in Richtung Maschine 1 in Bewegung setzt, sobald sich ein Teil $\text{A}[1]\text{Beladen}$ auf ihm befindet. Hier wird also abhängig vom Arbeitsplan bzw. dem Werkstück entschieden, in welcher Richtung (M1 oder M2) das FTS-Fahrzeug seinen Inhalt transportiert.

Damit sind bis auf die genannten Unterseiten der Arbeitsplanmodelle alle Teile der Beschreibung des Fertigungssystems dargestellt worden. Auf die noch fehlenden Bibliotheksmodule geht der folgende Abschnitt ein.

3.3 Eine Bibliothek von Teilmodellen

Um einmal definierte Teilmodelle mehrfach in ein Modell einsetzen zu können, wird eine entsprechende Bibliothek zur Verfügung gestellt. Ein solches Teilmodell sollte ein relativ abgeschlossenes Objekt eines realen Fertigungssystems sein, wie etwa eine Maschine oder ein Lager eines bestimmten Grundtyps. Fertigungssysteme sind dafür recht gut geeignet, da einige Typen von Bausteinen immer wieder mit unterschiedlichen Parametern auftauchen. Durch die Verwendung von vordefinierten Bibliotheksmodulen wird die Erstellung insbesondere von großen Modellen einfacher und weniger fehleranfällig.

Ein Bibliotheksmodul kann eine ganze Klasse einander ähnlicher Maschinen, Transporteinrichtungen o.ä. beschreiben, da durch verschiedene zusätzliche Konstrukte eine Art generisches Modell (*template*) erstellt wird. Die dafür verwendbaren Erweiterungen sind weiter unten erläutert. Zur Beschreibung eines solchen Teils eines Fertigungssystems gehört neben dem Strukturmodell auch eine generische Beschreibung der möglichen Arbeitspläne für dieses Strukturmodell. Diese werden wieder mit einem speziellen assoziierten Petri-Netz abgebildet. Struktur- und Arbeitsplanmodell eines Bibliotheksmoduls können aus mehreren hierarchisch angeordneten Seiten bestehen.

Hier soll keine erschöpfende Aufzählung von Teilmodellen angegeben werden. Vielmehr wird beschrieben, wie mit der entwickelten Modellierungsmethode Teilmodelle erstellt, parametrisiert und in ein größeres Modell eingebunden werden können. Es wird wieder das Beispiel aus 2.2 verwendet und Bibliotheksmodule für die zwei Maschinen, die vier Förderbänder sowie für die Montagestation des Beispiels gezeigt.

Neben der Erleichterung der Modellierung werden die vorhandenen Bibliotheksmodule außerdem dafür verwendet, sogenannte symbolische Modellobjekte zu erzeugen und um diese in ein spezielles Petri-Netz umzuwandeln. Mit dieser Thematik beschäftigt sich Kapitel 5.

Das für die beschriebene Modellierungsmethode entwickelte Softwarewerkzeug unterstützt die Verwaltung einer Bibliothek von Teilmodellen durch ihre Einteilung in Gruppen. Eine Gruppe kann Teilmodelle und andere Gruppen enthalten, wodurch eine Einteilung der Module in einer Baumstruktur möglich ist. Jede Gruppe steht für eine Menge von Bibliotheksmodulen mit ähnlichen Eigenschaften oder vergleichbarer Struktur. Damit wird das

Auffinden eines speziellen Moduls vereinfacht. Jedes Modul besteht dann wiederum aus dem generischen Modell der Struktur und dem des Arbeitsplans.

Genau wie bei den in Abschnitt 3.2 beschriebenen getrennten Modellen werden auch in den Bibliotheksmodellen im Strukturmodell die Eigenschaften der Betriebsmittel abgebildet, die unabhängig von den bearbeiteten Werkstücken sind. Das Arbeitsplanmodell enthält dementsprechend eine abstrakte Beschreibung der möglichen Fertigungsabläufe innerhalb des Strukturmodells. Es wird nur ein generisches Arbeitsplanmodell verwendet, weil sich aus diesem durch Verwendung von Existenzbedingungen und anderer Parameter (s.u.) beliebige Varianten eines Arbeitsplans abgeleitet werden können.

Die für die Teile der Bibliotheksmodule verwendeten Netzarten entsprechen im wesentlichen den auf den Seiten 39 (Struktur) und 43 (Arbeitspläne) beschriebenen. Um die Bibliotheksmodelle anpassen und mit Parametern instanziierten zu können, sind aber folgende Unterschiede und zusätzliche Konstrukte verfügbar:

- **Parameter** sind Objekte einer Bibliotheksseite, die einen Namen und einen Standardwert haben. Bei der Verwendung (Instanziierung) eines Bibliotheksmoduls kann der Modellierer einen Wert für den Parameter festlegen. Da bei der Instanziierung die Seite sichtbar ist, die über der neu eingefügten Unterseite liegt, beziehen sich alle Referenzen im Parameterausdruck auf Objekte dieser Oberseite. Parameter können genau wie Definitionen in den Ausdrücken auf der Bibliotheksseite verwendet werden. Über sie findet die Anpassung des generischen Modells an die speziellen Gegebenheiten des erzeugten Teilmodells statt.

Parametern können über die Benutzungsoberfläche Typen zugewiesen werden. Diese werden hauptsächlich für die spätere vereinfachte Eingabe von Werten und deren Überprüfung benötigt. Erlaubte Typen sind Ausdruck, Aufzählungstyp, Ganzzahlige Konstante, boolesche Konstante und in Arbeitsplanmodellen außerdem Eingangsmarke und Ausgangsmarke. Die beiden letzten Parametertypen werden für die symbolische Modellierung verwendet und im Kapitel 5 erklärt. Parameter werden in der grafischen Darstellung mit dem Schlüsselwort **Parameter** und dem jeweiligen Namen angegeben.

- Eine **Einschränkung** (*constraint*) besteht aus einem Namen und einem booleschen Ausdruck, der bei der Instanziierung des Bibliotheksmoduls ausgewertet wird. Ist das Ergebnis Falsch, wird der Vorgang abgebrochen und die zugeordnete Fehlermeldung ausgegeben. Damit können Einschränkungen des Wertebereichs von Parametern und ähnliches abgebildet werden. Sie werden in der grafischen Darstellung mit dem Schlüsselwort **Constraint** und ihrem Namen angegeben.
- Eine **Existenzbedingung** kann für Transitionen angegeben werden. Dies ist ebenfalls ein boolescher Ausdruck, der bei der Instanziierung des Bibliotheksmoduls ausgewertet wird. Die zugehörige Transition und ihre Kanten werden bei der Instanziierung aus dem Modell gelöscht, wenn der Ausdruck Falsch ergibt. Dadurch kann ein Bibliotheksmodul durch Parameter auch in der Struktur verändert werden. Für Stellen darf keine solche Bedingung angegeben werden, da sonst möglicherweise markierungsabhängige Ausdrücke fehlerhaft werden. Die Existenzbedingungen werden nicht in der grafischen Darstellung angezeigt.

- Da die oberste Seite eines Bibliotheksmoduls bei der Instanziierung die Unterseite einer neuen Vertretertransition wird, muß bekannt sein, welche Stellen dabei als Anschlußstellen in Frage kommen. Darum gibt es in Bibliotheksmodellen die Möglichkeit, Stellen der Modul-Primärseite als potentielle Anschlußstellen zu markieren. Diese haben vor der Instanziierung keine zugeordnete Socketstelle und werden wie normale Anschlußstellen gestrichelt gezeichnet. Für die Verwendung der Bibliotheksmodule in symbolischen Modellen haben die Anschlußstellen zwei zusätzliche Attribute: eine Eingangsbedingung und eine Ausgangsbedingung. Auf sie wird im Kapitel 5 eingegangen.
- An den Produktkanten der Arbeitspläne (nur solche sind dort erlaubt) kann in der Bibliothek ein Parameter statt einer Multimenge stehen. Dieser steht für eine Multimenge von Produktmarken, der bei der Instanziierung eingesetzt wird. Parameter dieser Art gehören — je nach Verwendung — zum Typ Eingangsmarke oder Ausgangsmarke. Ein- und Ausgang bezieht sich dabei jeweils auf die Richtung des Austauschs von Marken dieses Typs zwischen dem Teilmodell und der Umgebung. Jedes Bibliotheksmodul muß die Veränderung des Typs aller abfließenden Marken zulassen, auch wenn es sich um einen nicht zustandsändernden Vorgang wie einen Transport handelt. Ansonsten wäre es unmöglich, dieses Modul als letzten Arbeitsschritt in einer Fertigungsalternative zu verwenden, nach dem die Variante zurückgesetzt bzw. geändert werden muß.

Der Zustand eines Teilmodells kann von außen abgefragt werden, um z.B. Fertigungsalternativen abhängig von der Auslastung einer Fertigungszelle auszuwählen. Dazu sind die aus Namen und Ausdruck bestehenden Definitionen des Teilmodells außerhalb verwendbar. Bei der Erstellung eines Bibliotheksmodells sollte darum darauf geachtet werden, alle möglicherweise später interessanten inneren Zustände des Teilmodells über Definitionen verfügbar zu machen. Dies hat außerdem den Vorteil, daß eine Änderung des Ausdrucks der Definition von außen transparent ist. Sinnvolle Beispiele für solche Definitionen sind z.B. der Füllstand eines Lagers, der Intaktzustand einer Maschine oder der aktuelle Ort eines FTS-Fahrzeugs.

Bibliotheksmodul Förderband

Zunächst wird ein Modell für die Förderbänder des Beispiels erstellt. Bild 3.9 zeigt das entsprechende Strukturmodell. Wie in Kapitel 2.2 beschrieben, handelt es sich um ein kontinuierlich förderndes Band, auf dem sich die nicht entnommenen Werkstücke am Ziel stauen. Die Aufnahmefähigkeit des Bandes läßt sich aus der Bandlänge und der Größe der Werkstückträger ableiten.

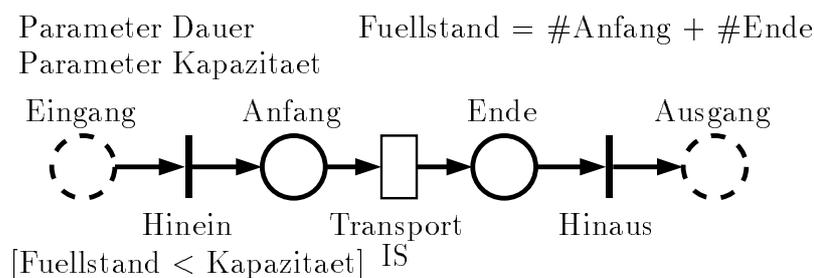


Abbildung 3.9: Strukturmodell eines Förderbandes

Die potentiellen Anschlußstellen sind in diesem Fall der **Eingang** und **Ausgang** des Bandes. Marken, die sich im **Eingang** befinden, werden sofort durch die zeitlose Transition **Hinein** auf den Bandanfang **Anfang** befördert, wenn die Kapazität des Bandes noch nicht erreicht ist. Dazu dient die Schaltbedingung $[Fuellstand < Kapazitaet]$. **Fuellstand** ist eine Definition und gibt die Anzahl der Marken auf dem Band (in den Stellen **Anfang** und **Ende**) an.

Die **Kapazitaet** des Bandes ist ebenso wie die **Dauer** (Durchlaufzeit eines Werkstückträgers durch das leere Förderband) ein Parameter des Bibliotheksmoduls. Dieses Modell läßt sich damit für alle Förderbänder dieser Art unabhängig von der Durchlaufzeit und der Kapazität verwenden. Der Parameter **Dauer** wird für die Schaltzeitverteilung der Transition **Transport** verwendet, die mit $Exponential(Dauer)$ angegeben ist. Der Standardwert der Kapazität ist eine und die Dauer fünf Zeiteinheiten. Die Transition **Transport** modelliert die Zeit, die jedes Teil benötigt, um zum Ende des Förderbandes zu gelangen. Da alle auf dem Förderband liegenden Teile gleichzeitig transportiert werden, hat die Transition die Schaltsemantik *infinite server* (IS).

Zur Vereinfachung des Modells wurde hier davon abstrahiert, daß sich Werkstückträger auf dem Förderband nicht überholen können. Diese FIFO-Eigenschaft hat das Modell nicht. Wenn es in einem Fall wichtig ist, dies mitzumodellieren, ist dies mit einer komplizierteren Beschreibung möglich. Da sich dabei aber auch der Zustandsraum stark vergrößert, müssen Vor- und Nachteile einer solchen Änderung abgewogen werden.

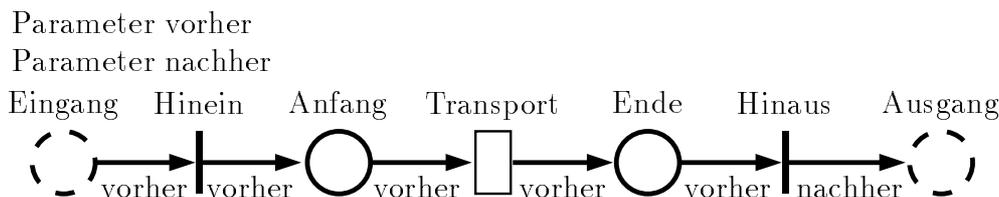


Abbildung 3.10: Arbeitsplanmodell des Förderbandes

Zum Strukturmodell des Förderbandes gehört das in Bild 3.10 dargestellte abstrakte Modell eines Arbeitsplans. Da es sich hier nur um einen Transport handelt, ändert sich der Typ des Werkstücks nicht. Trotzdem müssen zwei Parameter für den Markentyp (**vorher** und **nachher**) an den Kanten verwendet werden, da nach dem Transportvorgang eine Fertigungsalternative zu Ende sein kann und die Variantenbezeichnung sich dann ändert. Dies ist zum Beispiel bei der Verwendung des Förderband-Moduls in Bild 3.4 als Verfeinerung der Transitionen **Band1** und **Band2** der Fall. Die Instanziierung und Verwendung der Bibliotheksmodelle im Beispiel wird am Ende dieses Abschnitts erläutert.

Bibliotheksmodul einfache Maschine

Bild 3.11 zeigt das Modell der Struktur einer einfachen Maschine, die ausfallen und wieder repariert werden kann. Die Bearbeitung eines Werkstücks in der Stelle **BearbPlatz** erfolgt durch das Schalten der Transition **Bearbeiten**. Die dafür benötigte Zeit ist exponentiell verteilt mit dem Parameter **StandardZeit**, der ein Parameter des Bibliotheksmoduls ist.

Bearbeiten kann nur schalten, wenn die Schaltbedingung $[InOrdnung]$ erfüllt ist. **InOrdnung** ist als boolescher Ausdruck definiert, der wahr ist, wenn sich in der Stelle **POK** eine Marke

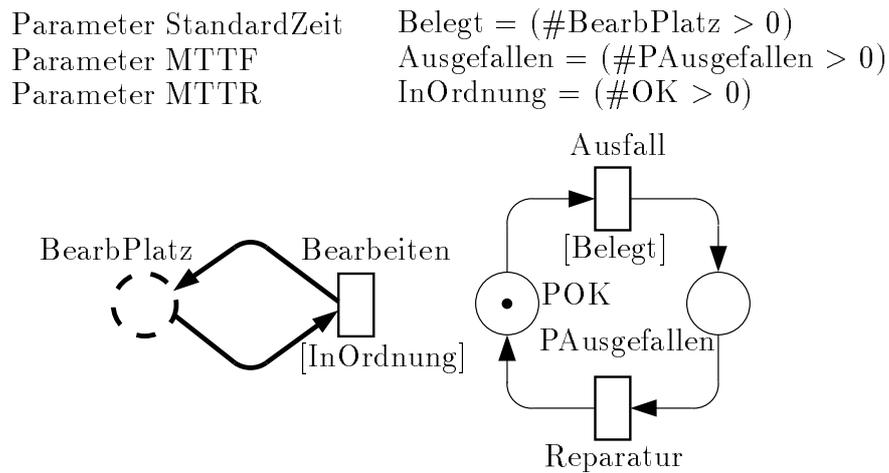


Abbildung 3.11: Strukturmodell einer Maschine mit Ausfällen

befindet. Der rechte, „farblose“ Teil des Modells bildet das Ausfall- und Reparaturverhalten der Maschine ab. Der aktuelle Zustand kann über die Stellen `POK` und `PAusgefallen` abgefragt werden. Die Transitionen `Ausfall` und `Reparatur` modellieren die entsprechenden Vorgänge. Ihre Schaltzeiten sind mit `Exponential(MTTF)` (*mean time to failure*, Transition `Ausfall`) und `Exponential(MTTR)` (*mean time to repair*, Transition `Reparatur`) angegeben. `MTTF` und `MTTR` sind ebenfalls Parameter des Modells. Ihre vorgegebenen Standardwerte seien 1200 und 60.

`Belegt` ist wahr, wenn sich ein Werkstück im Bearbeitungsplatz der Maschine befindet, und nur dann kann die Maschine ausfallen (`Ausfall` hat die Schaltbedingung `[Belegt]`). Die Definition von `Ausgefallen` wird hier nicht verwendet, darüber kann aber der Zustand des Teilmodells von außen abgefragt werden.

Das zugehörige Arbeitsplanmodell zeigt Bild 3.12. Um einen linearen Arbeitsplan zu erhalten, wurden die Kanten zwischen der Transition `Bearbeiten` und der Stelle `BearbPlatz` auseinandergezogen. Durch die Beschriftung der Kanten zwischen `BearbPlatz` und `Bearbeiten` mit den Parametern `vorher` und `nachher` wird die Veränderung im Bearbeitungszustand eines Werkstücks modelliert. Bei der Instanziierung wird später der Parameter `vorher` mit der Farbe des Ausgangszustands belegt, `nachher` entspricht dem Endzustand des Bearbeitungsschritts. Durch das Schalten der Transition `Bearbeiten` wird dann also eine Marke aus `BearbPlatz` abgezogen und im selben Schritt (atomares Schalten) eine neue Marke erzeugt. Über diesen Umweg wird der Zustand der Marke „verändert“. Die Schaltbedingung `[@BearbPlatz >= vorher]` an der Transition `Ausfall` bewirkt, daß die Maschine nur dann ausfallen kann, wenn sie gerade aktiv ist. Dies ist sie immer dann, wenn ein zu bearbeitendes Teil (`vorher`) sich in der Stelle `BearbPlatz` befindet. Der Vergleich `>=` entspricht hier

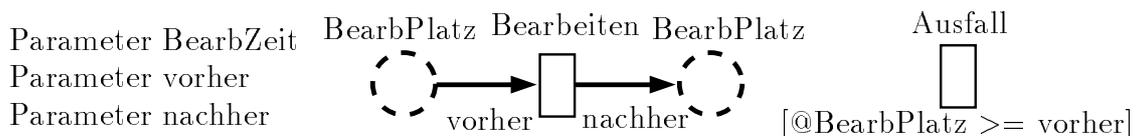


Abbildung 3.12: Arbeitsplanmodell für die Maschine

dem Test, ob eine Marke der Farbe `vorher` in der Stelle `BearbPlatz` enthalten ist (siehe Erklärung der Multimengen im Anhang B).

Als Parameter des Arbeitsplanmodells ist die Bearbeitungszeit für den speziellen Fertigungsschritt `BearbZeit` angebar, falls er sich von dem im Strukturmodell spezifizierten Standardwert unterscheidet. Außerdem sind die Parameter, die für die bearbeiteten Werkstücke stehen, als Parameter `vorher` und `nachher` angegeben.

Bibliotheksmodul Montagestation

Als letztes noch fehlendes Bibliotheksmodul wird nun die Montagestation erklärt, deren Strukturmodell Bild 3.13 zeigt. Die Montagestation ist mit der Umgebung über zwei Stellen verbunden: den Bearbeitungsplatz `PMontage`, auf dem sich das Hauptteil befindet, und den Eingang `EinZusatz`, über den die anzumontierenden Teile angeliefert werden. Die Transition `LadeZusatz` modelliert das Bewegen der Zusatzteile von `EinZusatz` zu einem Zwischenpuffer `PZusatz`, von dem sie dann durch die eigentliche Montagevorrichtung `Montieren` entnommen und montiert werden. Der Zwischenspeicher hat eine parametrisierbare Kapazität `Kap`, deren vorgegebener Wert 5 ist. Die Ladevorrichtung kann ausfallen und repariert werden. Dies wird wie auch schon bei der Maschine durch den rechten, „farblosen“ Modellteil abgebildet. Die Transition `LadeZusatz` kann darum nur schalten, wenn die Schaltbedingung `[LadeOK]`, die als $(\#POK > 0)$ definiert ist, wahr ergibt.

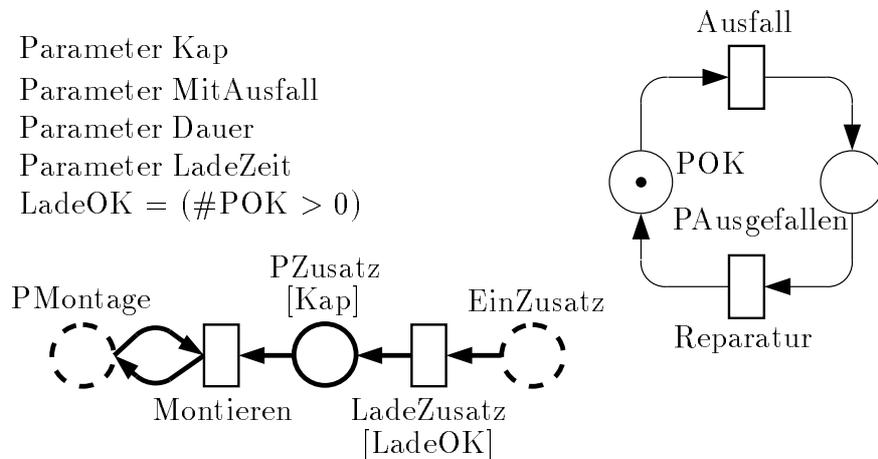


Abbildung 3.13: Strukturmodell der Montagestation

Der Modellierer soll später beim Instanzieren angeben können, ob die Ladevorrichtung ausfallen kann oder nicht. Dazu muß er nur dem Parameter `MitAusfall` den Wert `Wahr` oder `Falsch` zuordnen, es handelt sich also um einen booleschen Parameter. Den beiden Transitionen `Ausfall` und `Reparatur` ist jeweils die Existenzbedingung `MitAusfall` zugeordnet. Sie werden also später bei der Instanzierung nur dann ins Modell übernommen, wenn durch den Modellierer als Wert für den Parameter `MitAusfall` `Wahr` angegeben wurde. Existenzbedingungen können auch für kompliziertere Fälle verwendet werden, in denen sich die Teilmodelle sogar strukturell unterscheiden (mit oder ohne Vorpufferung etc.). Mit ihnen lassen sich auch mehrere unterschiedliche Transitionen, z.B. deterministische und exponentielle, für dieselbe

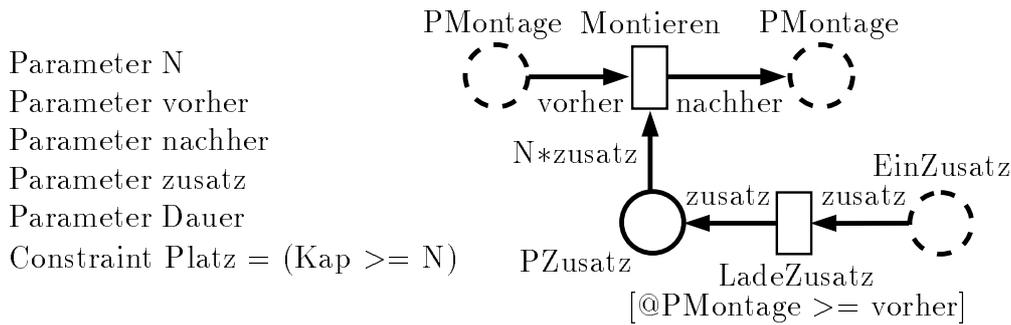


Abbildung 3.14: Arbeitsplanmodell der Montagestation

Aktivität parallel verwenden, wenn durch die Parametrisierung nur eine mit erfüllter Existenzbedingung übrigbleibt.

Über die Parameter `Dauer` und `LadeZeit`, deren voreingestellte Werte 10 bzw. 5 seien, kann die mittlere Zeit für einen Montagevorgang und das Laden eines Zusatzteils eingestellt werden. Für die Transitionen `Ausfall` und `Reparatur` wurden `Exponential(1000)` und `Exponential(60)` als Schaltzeitverteilungen gewählt.

Der dazu gehörende generische Arbeitsplan ist in Bild 3.14 dargestellt. Die Variablen `vorher` und `nachher` modellieren wieder die Veränderung des Bearbeitungszustands des Werkstücks in `PMontage`. Von `EinZusatz` werden Teile des Typs `zusatz` zur Stelle `PZusatz` bewegt. Beim eigentlichen Montagevorgang werden durch die Transition `Montieren` möglicherweise mehrere `zusatz`-Teile auf einmal montiert (`N*zusatz`). Diese Anzahl ist ein Parameter des Arbeitsplanmodells und ist mit `N` bezeichnet. Der Standardwert für `N` ist Eins.

Als weitere Parameter sind die Typen der Marken anzugeben, die im Teilmodell auftreten: `vorher` (Typ Eingangsmarke), `nachher` (Ausgangsmarke) und `zusatz` (Eingangsmarke).

Auf der Strukturseite dieses Bibliotheksmoduls war ja die Kapazität der Stelle `PZusatz` über den Parameter `Kap` angegeben worden. Falls bei der Verwendung des Moduls für einen Arbeitsplan `N` größer ist als `Kap`, werden nie ausreichend `zusatz`-Marken in `PZusatz` sein können, damit der Montagevorgang ausgeführt wird. Um einen solchen Spezifikationsfehler auszuschließen, wird die Einschränkung `Platz` mit der Bedingung `Kap >= N` eingeführt.

Die Ladevorrichtung `LadeZusatz` soll nur dann Zusatzteile in den Zwischenpuffer bringen, wenn sich ein zu montierendes Werkstück in `PMontage` befindet. Dies wird mit der Schaltbedingung `[@PMontage >= vorher]` erreicht.

Nachdem alle im Beispiel verwendeten Bibliotheksmodule beschrieben wurden, wird nun ihre Instanziierung erläutert. Dies wird im Zusammenhang mit Hinweisen für die Verwendung und Funktionsweise des Softwarewerkzeugs getan.

Instanziierung eines Strukturmodells

Entscheidet sich der Modellierer, im Beispiel auf der Strukturseite (Bild 3.2 auf Seite 40) eine Montagestation aus der Bibliothek einzufügen, ruft er einen entsprechenden Befehl („Insert Template“ aus dem Menu „Module“) der Benutzungsoberfläche auf. Danach muß er über ein Auswahlmenü das gewünschte Bibliotheksmodul bestimmen, wobei über die Einteilung in Gruppen das Auffinden erleichtert wird. Im Beispiel wird das in Bild 3.13 dargestellte Strukturmodell der Montagestation verwendet.

Dann fragt die Benutzungsoberfläche Werte für die Parameter der ausgewählten Strukturseite ab, also im Beispiel **Kap**, **MitAusfall**, **Dauer** und **Ladezeit**. Falls die voreingestellten Werte verwendet werden sollen, braucht nichts eingegeben zu werden. Je nach Typ des Parameters werden nur bestimmte Eingaben zugelassen, für eine ganzzahlige Konstante zum Beispiel keine Buchstaben. Zur Eingabe eines Aufzählungstyps kann ein *pull-down*-Menu verwendet werden, welches die erlaubten Werte enthält. Im Beispiel wird für **Kap** der Wert Vier und für **MitAusfall** Falsch ausgewählt; die Ladevorrichtung der Montagestation soll also hier nicht ausfallen können.

Wenn der Modellierer die Eingabe der Parameter beendet hat, beginnt die automatische Instanziierung des ausgewählten Bibliotheksmoduls. Zunächst müssen eventuell vorhandene Einschränkungen geprüft werden, was in diesem Fall nicht notwendig ist. Auf der aktuellen Seite wird dann eine Vertretertransition automatisch erzeugt, der zunächst eine leere Unterseite zugeordnet ist. Für die Vertretertransition wird vermerkt, mit welcher Bibliothekseite sie verfeinert wurde, um später den Bezug dazu herstellen zu können.

Dann werden die Objekte der Bibliotheksseite in die neue Unterseite kopiert. Dabei werden für die Parameter die angegebenen Werte eingesetzt. Für Transitionen muß außerdem getestet werden, ob ihre Existenzbedingung wahr ist. Im Beispiel werden die Transitionen **Ausfall** und **Reparatur** und die mit ihnen verbundenen Kanten nicht kopiert. **LadeOK** wird demnach immer Wahr sein.

Für alle als Anschlußstellen gekennzeichneten Stellen der Unterseite wird dann eine Sockelstelle auf der Oberseite erzeugt, die mit der Vertretertransition verbunden wird. Die Richtung der verbindenden Kante hängt von der Richtung der Kanten ab, die auf der Unterseite mit der Anschlußstelle verbunden sind. Die Instanziierung der Strukturseite ist damit vollständig.

Die automatisch erzeugten Stellen können entweder weiterverwendet oder von Hand mit den gewünschten Stellen verschmolzen werden. Dies geschieht in der Benutzungsoberfläche, indem eine Stelle auf eine andere gezogen wird. Die gezogene Stelle wird dabei gelöscht und alle ihre Kanten mit der anderen Stelle verbunden.

Instanziierung eines Arbeitsplanmodells

Normalerweise sollte das Arbeitsplanmodell nach dem Strukturmodell entworfen werden. Zum Beispiel werde auf der Primärseite des Arbeitsplans A (Bild 3.4 auf Seite 46) die Vertretertransition **Montage** erzeugt. Da sich diese Transition auf einer assoziierten Seite befindet, gehört sie zu der gleichnamigen Transition **Montage** auf der Primärseite des Strukturmodells. Weil sie bereits mit einem Bibliotheksmodul verfeinert wurde, wird auch für das Arbeitsplanmodell auf die Bibliothek zurückgegriffen. Dazu wurde bei **Montage** gespeichert, mit welchem Modul es verfeinert wurde, und der richtige generische Arbeitsplan (Bild 3.14) kann automatisch gefunden werden.

Die Instanziierung des Arbeitsplanmodells geschieht ähnlich wie oben beschrieben für das Strukturmodell. Zunächst werden Werte für die Parameter abgefragt: für **N** Zwei, da zwei Teile **C.Geliefert** verarbeitet werden sollen; der Wert von **zusatz** ist **C.Geliefert**, von **vorher** **A.Entgratet** und von **nachher** **A.Montiert**. Für **Dauer** und **LadeZeit** werden keine von den voreingestellten Werten abweichende Eingaben gemacht.

Bei der Angabe der Parameter, die die Ein- und Ausgangsmarken des Bibliotheksmoduls spezifizieren, ist eine Unterstützung durch die Benutzungsoberfläche möglich. Dabei müssen vor der Instanziierung die Kanten zu den umgebenden Stellen mit den entsprechenden Produkttypen hergestellt werden. Bei der nachfolgenden Instanziierung ist dann die Zuordnung der Ein- und Ausgangsstellen zum Teil eindeutig festgelegt. Im Beispiel ist **PMontage** die einzige Ausgangsstelle der Vertretertransition **Montage**; sie kann demnach nur zum Anschlußknoten **PMontage** der Bibliotheksseite gehören, da diese die einzige Ausgangsstelle des Bibliotheksmoduls ist. Dementsprechend muß der Parameter **nachher** der Unterseite mit dem Wert **A.Montiert** belegt werden. Auf diese Art kann ein Teil der Informationen aus der Modellstruktur abgeleitet werden, und nur die nicht eindeutigen Zuordnungen müssen noch manuell vorgenommen werden.

Nun beginnt die automatische Instanziierung des Bibliotheksmoduls. Die Einschränkung **Kap >= N** ist erfüllt, so daß der Vorgang nicht abgebrochen werden muß. Dann werden wieder die Objekte der Bibliotheksseite in eine neue Unterseite kopiert, die der Vertretertransition **Montage** zugeordnet ist. Dabei werden für die Parameter die angegebenen Werte eingesetzt. Die Schaltbedingung der Transition **LadeZusatz** wird dadurch [**@PMontage >= A.Entgratet**]. Da hier keine Existenzbedingungen angegeben wurden, können alle Transitionen kopiert werden.

Für die Anschlußstellen **PMontage** und **EinZusatz** der Unterseite wird jeweils eine Sockelstelle auf der Oberseite erzeugt und mit der Vertretertransition verbunden. Die Instanziierung der Arbeitsplanseite ist damit vollständig. Dieser Vorgang muß für jedes Auftreten der Transition **Montage** in einem Arbeitsplanmodell wiederholt werden. Im Beispiel ist das nur ein weiteres Mal im Arbeitsplan B notwendig, und zwar für **vorher** gleich **B.OK** und **nachher** **B.Montiert**.

Auf dieselbe Art und Weise müssen die Verfeinerungen der Maschinen und der Förderbänder im Struktur- und Arbeitsplanmodell vorgenommen werden. Am dabei entstehenden vollständigen Modell des Fertigungssystems und seiner Arbeitspläne sind nun noch manuelle Änderungen möglich, bevor daraus ein Gesamtmodell zur anschließenden Untersuchung generiert wird.

3.4 Automatische Erzeugung eines Gesamtmodells

In den vorherigen Abschnitten wurde gezeigt, wie man jeweils ein Modell eines Fertigungssystems und der Arbeitspläne als spezielles Petri-Netz erzeugt. Nun muß aus diesen Teilen ein Gesamtmodell generiert werden, welches das Verhalten des Fertigungssystems bei der Produktion der beschriebenen Werkstücke spezifiziert. Eine Methode zur automatischen Umwandlung in ein Gesamtmodell wird beschrieben. Dabei bleibt das Modell der Struktur des Fertigungssystems erhalten, was der Verständlichkeit dient.

Nach der Umwandlung von Alternativen mit Hilfe von Entscheidungstransitionen werden zunächst sogenannte teilweise Übergangstabellen aus jedem Arbeitsplanmodell und dem Strukturmodell erzeugt. Daraus wird für jede Transition des Strukturmodells eine vollständige Übergangstabelle generiert. In einer Übergangstabelle sind die Schaltmöglichkeiten einer Transition und alle dazugehörigen Eigenschaften enthalten. Sie ersetzt die sonst in farbigen Petri-Netzen nötigen Schaltprädikate und Kantenausdrücke. Die Tabellen beschreiben dann zusammen mit dem Strukturmodell das gesamte Verhalten. Das dabei entstehende Gesamtmodell entspricht der in Abschnitt 3.1.1 angegebenen Definition. Letztlich werden durch diesen Vorgang die allgemeinen strukturellen Einschränkungen und Standardwerte des Fertigungssystems mit den arbeitsplanabhängigen Eigenschaften der Betriebsmittel verknüpft.

Während der Modellerstellung kann das Modell in einem unfertigen Zustand sein, der noch nicht den einzelnen Bedingungen genügt. Bevor aus einem Modell ein Gesamtmodell erzeugt werden kann, ist darum die Durchführung eines Konsistenztests durch die verwendete Benutzungsoberfläche sinnvoll. Dabei sollten die Eigenschaften überprüft werden, die in den Besonderheiten der verschiedenen Netzklassen genannt sind.

Fertigungsvarianten und Entscheidungstransitionen

Eine Fertigungsalternative oder -variante ist daran erkennbar, daß von einer Stelle ausgehend verschiedene Wege bzw. Kanten beschriftet werden können. Um die Eindeutigkeit des Arbeitsplans aufrechtzuerhalten, müssen die Folgekanten der Stelle unterschiedlich gekennzeichnet sein. Da sich aber die Werkstücke im selben Bearbeitungszustand befinden, kann eine Unterscheidung nur über die Variante vorgenommen werden, die aus diesem Grund eingeführt wurde.

Mit der Entscheidung über die zu wählende Alternative für ein Werkstück wird der entsprechenden Marke die dazugehörige Variante zugewiesen, also der Markentyp verändert. Ein solcher Vorgang ist am natürlichsten mit einer zeitlosen Transition modellierbar, da er ohne Zeitverzug abläuft. Die bei der Alternative im Arbeitsplan angebbaren Bedingungen und Wahrscheinlichkeiten lassen sich so leicht in das Modell integrieren. Darum müssen bei der Erzeugung des Gesamtmodells sogenannte **Entscheidungstransitionen** zusätzlich in das Modell eingefügt werden, die einer Marke die ausgewählte Variante aufprägen, sie aber am selben Ort belassen. Damit wird dem Modellierer die Arbeit abgenommen, sie von Hand einzutragen.

Die in einem Arbeitsplanmodell zulässige Angabe von Alternativen (siehe Bild 3.15 links) ist also letztlich nur eine verkürzte Schreibweise für das rechts im selben Bild gezeigte Teilmodell. Dabei stehen [Bed1] und [Bed2] für zwei Bedingungen, **Wkt1** und **Wkt2** für zwei

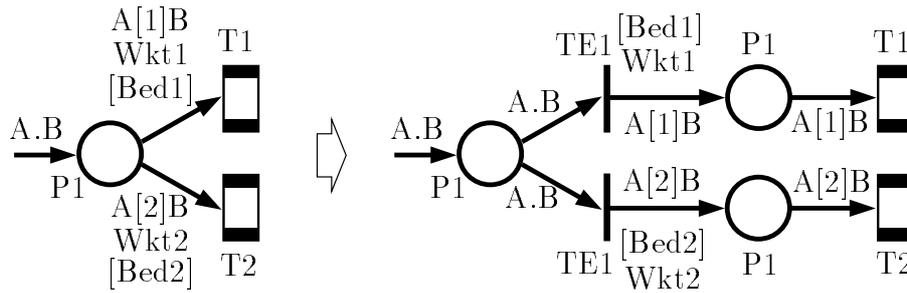


Abbildung 3.15: Umwandlung von Fertigungsalternativen im Arbeitsplanmodell

Wahrscheinlichkeiten. TE1 ist der Name der erzeugten Entscheidungstransition, an der jeweils die Schaltbedingung und die Wahrscheinlichkeit angegeben ist. Diese Vorgehensweise läßt sich genauso auf eine beliebige Anzahl von Alternativen erweitern, für jede Möglichkeit muß dann eine Transition und eine Stelle eingefügt werden. Bei der Erzeugung eines Gesamtmodells wird die erste in die zweite Darstellungsart umgewandelt, bevor das eigentliche Gesamtmodell berechnet wird.

Dabei müssen die Entscheidungstransitionen außer im Arbeitsplanmodell auch im Strukturmodell erzeugt und ihnen beliebige, noch nicht auftretende Namen zugewiesen werden (siehe Bilder 3.15 und 3.16). Diese Vorgehensweise bei der Umwandlung von Alternativen im Arbeitsplan ist nur eine Erleichterung bei der Modellierung. Das gewünschte Verhalten kann auch direkt von Hand spezifiziert werden, ohne die Entscheidungstransitionen automatisch erzeugen zu lassen. Dann muß der Modellierer selbst Konstrukte ähnlich der sonst automatisch erzeugten Entscheidungstransitionen einfügen. Die Wahrscheinlichkeiten und Bedingungen können dann manuell an den zusätzlichen Transitionen eingefügt werden.

Nach dem Ende einer Fertigungsalternative können die Varianten wieder in eine Stelle münden (siehe z.B. Bild 3.4). In diesem Fall sind die Bearbeitungszustände der Werkstücke aus den Varianten gleich (hier A.Loeh2), und sie müssen nicht mehr unterschieden werden. Der Typ der Marken ist darum schon nach dem Ende des letzten Bearbeitungsschritts jeder Variante identisch. Dies entspricht den Regeln für Arbeitspläne, nach denen alle in einer Stelle endenden Kanten denselben Markentyp transportieren müssen.

Mit den automatisch erzeugten Entscheidungstransitionen ist bekannt, an welchen Stellen im Modell Entscheidungen über Fertigungsalternativen getroffen werden. Sie könnten demnach als Schnittstelle zwischen dem Modell und von außen angebbaren Steueranweisungen dienen. Zum Beispiel könnte ein Planungsalgorithmus statt der Bedingungen an den Transitionen eingesetzt werden, um seine Auswirkungen anhand des Modells überprüfen zu können.

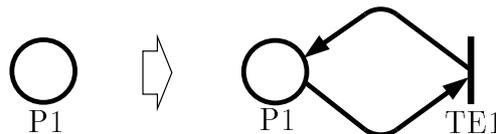


Abbildung 3.16: Umwandlung von Fertigungsalternativen im Strukturmodell

Ein Beispiel

Die Erzeugung des Gesamtmodells wird hier weiter an dem bereits in den vorangegangenen Abschnitten verwendeten Beispiel gezeigt. Aus Platzgründen werden nur einige interessante Teile des Modells herausgegriffen, und zwar die Transitionen **ABLiefer** und **Handarbeit** der Primärseite, **LM1**, **LM2** und **M2L** aus der Unterseite des fahrerlosen Transportsystems sowie die für die zwei auftretenden Alternativen automatisch erzeugten Entscheidungstransitionen **TE1** und **TE2**.

Der erste Schritt bei der Erzeugung eines Gesamtmodells ist wie oben beschrieben die Ersetzung der Alternativen durch Entscheidungstransitionen. Dies ist in unserem Beispiel an zwei Stellen notwendig. Im Arbeitsplan A (Bild 3.4) existieren zwei Alternativen für Werkstücke des Typs **A.Beladen** in Stelle **PLade**. Demnach muß eine zeitlose Entscheidungstransition an der Stelle **PLade** in der Struktur eingefügt werden, und diese im Arbeitsplanmodell nach der oben gezeigten Methode verwendet werden. In diesem Fall sind keine Wahrscheinlichkeiten angegeben, sondern nur Bedingungen, so daß die Wahrscheinlichkeiten auf den Wert Eins gesetzt werden. Die zweite Entscheidungstransition muß aufgrund der verschiedenen Prüfergebnisse im Arbeitsplan B (Bild 3.5) nach der Stelle **PHand** eingefügt werden. Dies geschieht genau wie im ersten Fall, nur daß hier keine Bedingungen, aber dafür Wahrscheinlichkeiten angegeben sind. Für die Bedingungen wird darum der Wert **Wahr** eingesetzt. Es wird angenommen, daß die eingefügten Transitionen die Namen **TE1** und **TE2** erhalten.

Im zweiten Schritt in Richtung Gesamtmodell werden die Struktur- und die Arbeitsplanmodelle untersucht und sogenannte **teilweise Übergangstabellen** generiert. Diese enthalten jeweils Teilstücke der Betriebsmittelbeschreibung und werden später zusammengesetzt. Vertretertransitionen können keine Übergangstabellen besitzen, da sie ja nur Platzhalter für Unterseiten und keine echten Transitionen sind. Der inhaltliche Aufbau einer Übergangstabelle wurde in der Definition in Abschnitt 3.1.1 dargestellt. Hier wird kurz an den ausgewählten Beispielen der Aufbau gezeigt, die genaue Syntax enthält Anhang C. Eine Übergangstabelle wird durch eckige Klammern begrenzt und besteht aus einer Menge von Einträgen, die jeweils eine Zeile der Tabelle bilden:

$$[(u^{guard}) u^{in}, u^{out}; u^{select}; u^{weight}; u^{delay}]$$

Das Ein- und Ausgangsverhalten u^{in} bzw. u^{out} wird durch Angabe der Stelle und der Multimenge von Marken, die von oder zu dieser Stelle bewegt werden, angegeben. Leere Felder können weggelassen werden, aber die Trennzeichen $;$ müssen für die eindeutige Zuordnung erhalten bleiben.

Nun werden die teilweisen Übergangstabellen erzeugt. Jedes Auftreten einer Transition in einem Arbeitsplanmodell wird in einen entsprechenden Tabelleneintrag umgewandelt:

- Die Schaltbedingung u^{guard} ergibt sich aus der Schaltbedingung der Transition im Arbeitsplanmodell.
- Das Ein- und Ausgangsschaltverhalten wird aus den mit der Transition durch Kanten verbundenen Stellen im Arbeitsplanmodell sowie den an den Kanten stehenden Multimengen von Farben abgeleitet.
- Falls es sich um eine zeitbehaftete Transition mit der Schaltsemantik *exclusive server* handelt, wird die Auswahlwahrscheinlichkeit des Übergangs u^{select} übernommen.

- Ebenso werden Schaltgewicht u^{weight} und Schaltzeitverteilung u^{delay} aus dem Arbeitsplanmodell in die Übergangstabelle kopiert.

Für die ausgewählten Transitionen ergeben sich folgende Einträge aus dem Arbeitsplan A:

```
LM1      [(@AufFTS=A[1]Beladen);;;]
LM2      [(@AufFTS=A[2]Beladen);;;]
ABLiefer [out#LagerAB=A.Geliefert; 0.7; ;]
Handarbeit [in#PHand=A.Loeh2, out#PHand=A.Entgratet;;; Exponential(40)]
TE1      [(!M1\Belegt)in#PLade=A.Beladen,out#PLade=A[1]Beladen;;; Immediate]
TE1      [(M1\Belegt) in#PLade=A.Beladen,out#PLade=A[2]Beladen;;; Immediate]
```

Für Arbeitsplan B ergibt sich folgendes:

```
LM1      [(@AufFTS=B.Beladen);;;]
LM1      [(@AufFTS=B.Nachbohr);;;]
ABLiefer [out#LagerAB=B.Geliefert; 0.3; ;]
Handarbeit [in#PHand=B[1]GLoeh, out#PHand=B.OK;;; Exponential(20)]
Handarbeit [in#PHand=B[2]GLoeh, out#PHand=B.Nachbohr;;; Exponential(20)]
TE2      [in#PHand=B.GLoeh, out#PHand=B[1]GLoeh;;; 0.95; Immediate]
TE2      [in#PHand=B.GLoeh, out#PHand=B[2]GLoeh;;; 0.05; Immediate]
```

Die Transition LM2 kommt außerdem noch im Arbeitsplan der Werkstückträger vor:

```
LM2      [(@AufFTS=WT.leer);;;]
```

Als nächstes werden auch aus dem Strukturmodell teilweise Übergangstabellen generiert, in denen die Standardattribute der Transitionen enthalten sind. Für jede Transition des Strukturmodells wird eine Tabelle mit genau einem Tabelleneintrag erzeugt:

- Die Schaltbedingung u^{guard} kann weggelassen werden, da sie hier keine Rolle spielt.
- Das Ein- und Ausgangsschaltverhalten wird aus den mit der Transition durch Kanten verbundenen Stellen im Arbeitsplanmodell abgeleitet. Dabei werden ausschließlich elementare Kanten berücksichtigt, denn Produktkanten sind nur potentielle Verbindungen mit Stellen und werden nur durch ihre Verwendung in einem Arbeitsplanmodell aktiviert.
- Die Auswahlwahrscheinlichkeit des Übergangs u^{select} hat hier keine Bedeutung.
- Schaltgewicht u^{weight} und Schaltzeitverteilung u^{delay} werden aus dem Modell in die Übergangstabelle kopiert.

Die teilweisen Übergangstabellen aus dem Strukturmodell ergeben sich demnach wie folgt:

```
LM1      [in#BeiL=e, out#BeiM1=e; 1; 1; Exponential(10)]
LM2      [in#BeiL=e, out#BeiM2=e; 1; 1; Exponential(12)]
M2L      [in#BeiM2=e, out#BeiL=e; 1; 1; Exponential(12)]
ABLiefer [; 1; 1; Exponential(30)]
Handarbeit [; 1; 1; Exponential(60)]
TE1      [; 1; 1; Immediate]
TE2      [; 1; 1; Immediate]
```

Im letzten Schritt werden die teilweisen Übergangstabellen zusammengefaßt und den Transitionen des Strukturmodells zugeordnet. Die Arbeitsplanmodelle sind dann nicht mehr im Gesamtmodell enthalten, da die in ihnen enthaltene Information jetzt in den Übergangstabellen versteckt ist. Das Zusammenfügen der Übergangstabellen geschieht, indem für jede Transition alle Einträge in teilweisen Übergangstabellen gesucht und zusammen oder-verknüpft (gekennzeichnet durch das Zeichen |) in eine Tabelle eingetragen werden. Diese wird dann als **vollständige Übergangstabelle** der Transition bezeichnet.

Bei der Zusammenstellung der Einträge werden für alle fehlenden Werte die entsprechenden Daten aus der teilweisen Übergangstabelle der Struktur übernommen, da diese Standardwerte darstellen. Ist eine Transition durch Kanten mit elementaren Stellen verbunden, wird aus dieser Übergangstabelle außerdem das Ein- und Ausgangsverhalten zu diesen Stellen in jeden Übergang der Transition zusätzlich übernommen.

Einen Sonderfall stellen Transitionen dar, die ausschließlich mit elementaren Stellen verbunden sind und in keinem Arbeitsplan auftauchen wie M2L im Beispiel. Diese Transitionen modellieren Vorgänge, die unabhängig von allen Arbeitsplänen stattfinden, und müssen mit ihrem durch die Struktur vorgegebenen Verhalten in das Gesamtmodell übernommen werden. Sie erhalten deshalb eine Übergangstabelle, die genau aus den Daten der teilweisen Übergangstabelle der Struktur stammen. Für alle anderen Transitionen dagegen gilt: wenn sie nicht mindestens in einem Arbeitsplan verwendet werden, haben sie eine leere Übergangstabelle und können damit nicht aktiviert werden bzw. schalten.

Die vollständigen Übergangstabellen der Transitionen lauten dann:

```
LM1      [(@AufFTS=A[1]Beladen) in#BeiL=e,out#BeiM1=e;1;1;Exponential(10)|
          (@AufFTS=B.Beladen)   in#BeiL=e,out#BeiM1=e;1;1;Exponential(10)|
          (@AufFTS=B.Nachbohr)  in#BeiL=e,out#BeiM1=e;1;1;Exponential(10)]

LM2      [(@AufFTS=A[2]Beladen) in#BeiL=e,out#BeiM2=e;1;1;Exponential(12)|
          (@AufFTS=WT.leer)     in#BeiL=e,out#BeiM2=e;1;1;Exponential(12)]

M2L      [in#BeiM2=e,out#BeiL=e;1;1;Exponential(12)]

ABLiefer [out#LagerAB=A.Geliefert;0.7;1;Exponential(30)|
          out#LagerAB=B.Geliefert;0.3;1;Exponential(30)]
```

```

Handarbeit [in#PHand=A.Loch2, out#PHand=A.Entgratet;1;1;Exponential(40)|
           in#PHand=B[1]GLoch,out#PHand=B.OK;          1;1;Exponential(20)|
           in#PHand=B[2]GLoch,out#PHand=B.Nachbohr; 1;1;Exponential(20)]

TE1      [(!M1\Belegt)in#PLade=A.Beladen,out#PLade=A[1]Beladen;1;1;Immediate|
          (M1\Belegt) in#PLade=A.Beladen,out#PLade=A[2]Beladen;1;1;Immediate]

TE2      [in#PHand=B.GLoch,out#PHand=B[1]GLoch;1;0.95;Immediate|
          in#PHand=B.GLoch,out#PHand=B[2]GLoch;1;0.05;Immediate]

```

Die Übergangstabelle einer Transition enthält nun die vollständige Beschreibung ihres Schaltverhaltens. Dies gilt auch für Transitionen, die mit Elementarstellen verbunden sind. Bei der Untersuchung des Verhaltens muß dann nur noch die Übergangstabelle beachtet werden, nicht mehr die Kanten im grafischen Modell, da deren Informationen vollständig in den Übergangstabellen enthalten sind.

Alle Informationen, die nicht die Übergangstabellen der Transitionen betreffen, werden im Gesamtmodell aus dem Strukturmodell übernommen. Dazu gehören die Stellenattribute, die Schaltsemantik und die Schaltbedingungen der Transitionen (z.B. [Leer] für M2L). Die gezeigten Schritte, die zu einem Gesamtmodell führen, sind automatisch ohne Nutzerinteraktion ausführbar. Sie ergeben ein Modell, welches das Verhalten des Fertigungssystems bei der Herstellung der in den Arbeitsplänen beschriebenen Produkten beschreibt. Dieses Modell entspricht äußerlich dem Strukturmodell, welches um versteckte Informationen erweitert wurde, und kann nun untersucht werden.

Kapitel 4

Bewertungsverfahren

Nachdem im letzten Kapitel eine spezielle Modellierungstechnik für Fertigungssysteme eingeführt wurde, enthalten die folgenden Abschnitte Techniken zur Ableitung von qualitativen und quantitativen Aussagen aus einem Modell. Dabei wird gezeigt, wie sich bekannte Methoden auf die Modelle anwenden lassen und welche Änderungen dadurch notwendig sind. Einige der angewandten Untersuchungsmethoden vereinfachen sich bzw. sind in dieser Form möglich, weil die Einschränkungen der speziellen Modellierungsart ausgenutzt werden.

4.1 Fragestellungen und Leistungsmaße

Bei der Erstellung eines Modells eines Fertigungssystems und seiner nachfolgenden Bewertung treten verschiedenartige Problemstellungen auf. Zunächst muß überprüft werden, ob das Modell das reale System so abbildet, daß die relevanten Eigenschaften — und möglichst nur diese — wiedergegeben werden. Anschließend können grundsätzliche Fragen über die Funktionsweise des modellierten Systems beantwortet werden, die meist mit Ja oder Nein zu beantworten sind. Diese qualitativen Fragestellungen können in solche unterteilt werden, die anhand der Modellstruktur beantwortbar sind und hier als strukturelle bzw. statische Eigenschaften bezeichnet werden. Im Gegensatz dazu gibt es qualitative Eigenschaften, die nur anhand des Erreichbarkeitsgraphen überprüft werden können und dynamische Eigenschaften genannt werden. Dazu zählt insbesondere die Lebendigkeit des Modells, die gewährleistet, daß das spezifizierte Fertigungssystem nie zum Stillstand z.B. aufgrund einer Verklemmung kommt. Mit Hilfe des Markenspiels kann das Verhalten interaktiv simuliert werden (*token game*), um ein besseres Verständnis des Modells zu erlangen oder dessen Funktionsweise zu überprüfen. Im ersten Unterabschnitt dieses Kapitels wird auf die Bestimmung einiger qualitativer Eigenschaften eingegangen.

Im Anschluß kann eine Leistungsbewertung des modellierten Fertigungssystems erfolgen, worauf in dieser Arbeit der Schwerpunkt liegt. Die für die Bewertung der Leistungsfähigkeit und Zuverlässigkeit eines Fertigungssystems wichtigsten Aussagen sind quantitativer Art. Sie geben z.B. Auskunft über den Durchsatz, die Auslastung oder die Verfügbarkeit und werden hier als Leistungsparameter bezeichnet. Ihre Bestimmung kann mit Hilfe von numerischer Analyse oder Simulation erfolgen, wie in den entsprechenden Abschnitten dieses

Kapitels beschrieben. Mit dem Vergleich der Leistungsparameter für verschiedene Auslegungen oder Arbeitsplanvariationen kann z.B. die Produktivität erhöht oder die Pufferfüllstände verringert werden.

Für die Spezifikation und Berechnung von Leistungsmaßen sind verschiedene Vorschläge bekannt [2, 19]. Die Beschreibung der Lösungsalgorithmen in dieser Arbeit beschränkt sich auf die aus der Wahrscheinlichkeitsverteilung der Modellzustände im eingeschwungenen (stationären) Zustand berechenbaren Leistungsparameter (*rate rewards* nach [19]). Darunter sind Aussagen über das modellierte System zu verstehen, die nach einer genügend langen bzw. unendlichen Einschwingzeit des Modells gemacht werden. Daraus ableitbar sind außerdem Aussagen über den Durchsatz von Transitionen, wenn diese eine exponentielle Schaltzeit haben oder nicht durch andere Übergänge deaktivierbar sind. Falls Aussagen über den Systemzustand zu einem speziellen Zeitpunkt z.B. nach Anlauf der Produktion benötigt werden, kann die transiente Analyse angewendet werden [17, 30, 31, 37], die hier aber nicht näher beschrieben wird.

Leistungsparameter gehören zur Spezifikation eines analysierbaren Modells, wurden aber nicht in die Definition im letzten Kapitel aufgenommen. Ein Parameter wird stets auf einer bestimmten Seite eines speziellen Petri-Netzes angegeben und besteht aus dem Namen, einem Ausdruck und einem Wert. Der Wert ist zunächst undefiniert und wird nach der Auswertung ersetzt, nachdem der Ausdruck für die Lösung des Modells berechnet wurde. Der einem Leistungsparameter zugeordnete und zu berechnende Ausdruck besteht aus miteinander verknüpften Grundtermen, mit denen sich u.a. Mittelwerte von Ausdrücken und Wahrscheinlichkeiten von Zuständen ausdrücken lassen. Dabei können folgende Konstrukte verwendet werden, wobei $expr^{\mathbf{B}}$ und $expr^{\mathbf{R}}$ für markierungsabhängige boolesche bzw. reellwertige Ausdrücke stehen:

- $P[expr^{\mathbf{B}}]$ Für die Wahrscheinlichkeit eines booleschen Ausdrucks $expr^{\mathbf{B}}$.
- $P[expr_1^{\mathbf{B}} | expr_2^{\mathbf{B}}]$ Wahrscheinlichkeit eines booleschen Ausdrucks $expr_1^{\mathbf{B}}$ unter der Bedingung, daß Ausdruck $expr_2^{\mathbf{B}}$ gilt (bedingte Wahrscheinlichkeit).
- $E[expr^{\mathbf{R}}]$ Erwartungswert eines reellwertigen Ausdrucks $expr^{\mathbf{R}}$.
- $E[expr^{\mathbf{R}} | expr^{\mathbf{B}}]$ Erwartungswert eines reellwertigen Ausdrucks $expr^{\mathbf{R}}$, wobei nur Zustände beachtet werden, in denen der boolesche Ausdruck $expr^{\mathbf{B}}$ gilt.
- $TP[t]$ Der Durchsatz einer zeitbehafteten Transition t . Dieser Wert gibt die durchschnittliche Anzahl des Schaltens der Transition pro Zeiteinheit des Modells an.

Beispiele für die Anwendung dieser Ausdrücke finden sich für das Anwendungsbeispiel im Kapitel 6. Häufig auftretende Fragestellungen können dabei wie folgt spezifiziert werden:

- Die Auslastung einer Maschine kann über die Wahrscheinlichkeit dafür berechnet werden, daß sich in der Bearbeitungsstelle mindestens eine Marke eines zu bearbeitenden Typs befindet. Über die getrennte Abfrage für verschiedene Markentypen kann die Belegung der Maschine mit bestimmten Bearbeitungsaufgaben bestimmt werden.
- Die Wahrscheinlichkeit dafür, daß eine Maschine ausgefallen ist, kann über die Markierung der Stelle berechnet werden, die den Fehlerzustand der Maschine modelliert.

- Die Zeitanteile, in denen eine Maschine blockiert ist oder durch fehlenden Nachschub warten muß, können über die Wahrscheinlichkeiten der entsprechenden Zustände ausgedrückt werden.
- Der Durchsatz eines Fertigungssystems, also die mittlere Anzahl an fertiggestellten Produkten, kann z.B. über den Durchsatz einer Ausgangstransition ermittelt werden.
- Der mittlere Füllstand eines Lagers oder Zwischenpuffers kann über den Erwartungswert der Markenanzahl in der Stelle abgefragt werden. Dies ist auch nach Produktarten getrennt möglich.
- Die Aufteilung der Verweildauer eines Produktes im Fertigungssystem in Bearbeitung, Warten und Transport ist über die jeweiligen Wahrscheinlichkeiten ausdrückbar, mit denen sich ein solches Teil in einer der entsprechenden Stellen befindet.
- Engpässe im modellierten Fertigungssystem können an hohen Pufferfüllständen davor oder an einer überdurchschnittlichen Auslastung erkannt werden.

4.2 Qualitative Eigenschaften und ihre Bestimmung

Dieses Kapitel geht auf die Untersuchung qualitativer Eigenschaften eines Fertigungssystems anhand eines speziellen Petri-Netzes ein. Der Vorteil dieser Untersuchungsart liegt darin, daß viele Aussagen unabhängig von der Markierung ausschließlich aus der Netzstruktur gewonnen werden können. Diese werden als strukturelle Eigenschaften eines Modells bezeichnet, und für ihre Bestimmung sind keine langen Berechnungszeiten erforderlich. Strukturelle Aussagen sind in den ersten Phasen des Entwurfs und der Modellierung hilfreich, um grundlegende Fehler zu entdecken und zu beheben. Die mögliche Ableitung von Ergebnissen aus der Modellstruktur ist ein Vorteil von Petri-Netzen gegenüber Simulationssprachen. Im weiteren werden einige Eigenschaften und Methoden zu ihrer Bestimmung für spezielle farbige Petri-Netze dargestellt. Eine umfassendere Darstellung struktureller Eigenschaften kann z.B. [59, 62, 72] entnommen werden.

Bei der Untersuchung der Netzstruktur spielt der später für die Analyse des Erreichbarkeitsgraphen notwendige erweiterte Zustandsbegriff keine Rolle. Darum wird hier vereinfachend nur von Markierungen gesprochen, zwischen denen es Zustandsübergänge durch das Schalten von Übergängen geben kann. Dieses Verhalten wird nicht formal eingeführt, entspricht aber dem natürlichen Ansatz, daß das Schalten eines Übergangs die im Eingangsverhalten u^{in} angegebenen Marken von den Eingangsstellen entfernt und die im Ausgangsverhalten u^{out} angegebenen Marken in den Ausgangsstellen erzeugt werden.

Stellen-Invarianten

Eine Stellen-Invariante ist allgemein eine Funktion über Stelleninhalten, die in jeder aus einer bestimmten Anfangsmarkierung heraus erreichbaren Markierung das gleiche Ergebnis liefert. Der Wert der Funktion ist demnach invariant gegenüber dem Schalten beliebiger Transitionen des Modells. Wird die Funktion auf die Anfangsmarkierung angewendet, ergibt sich der

Wert der Invariante. Daraus läßt sich eine Gleichung der Form $Funktion(Markierung) = Wert$ ableiten, die als Markierungsinvariante (*marking invariant*) bezeichnet wird. Aus der umfangreichen Literatur zu diesem Thema sei hier nur auf [39, 60, 23] verwiesen.

Invarianten stellen eine Möglichkeit zum Nachweis grundlegender struktureller Eigenschaften eines Modells dar. Ihre Berechnung erfordert nicht die Untersuchung des Erreichbarkeitsgraphen, sondern nur der Netzstruktur. Durch die Überprüfung der Invarianten eines Modells sind frühzeitige Korrekturen an fehlerhaft spezifizierten Modellen möglich. Sie werden außerdem benötigt, um dynamische qualitative Eigenschaften wie die Lebendigkeit oder Begrenztheit zu überprüfen. Ihre Interpretation für einige Eigenschaften von Fertigungssystemen wird am Ende dieses Abschnitts beschrieben.

In der Literatur werden zwei Methoden zur Untersuchung der Invarianten eines Modells verwendet [40]. Bei der interaktiven Bestimmung müssen vermutete Invarianten vom Modellierer angegeben werden, die dann anhand der Netzstruktur überprüft werden. Diese Methode ist einfacher zu realisieren, erfordert aber vom Modellierer weitgehende Kenntnisse über Invarianten. Da es Ziel der vorliegenden Arbeit ist, auch Modellierern ohne größeres Hintergrundwissen Methoden zur Verfügung zu stellen, wird statt dessen die automatische Berechnung verwendet. Diese hat im Gegensatz zur interaktiven Methode den Nachteil, daß sie komplexere Algorithmen erfordert und oft zu einer großen Menge schwer zu interpretierenden Invarianten führt. Um diese Probleme zu umgehen, lassen sich die Einschränkungen in der verwendeten Netzklasse ausnutzen.

Da normalerweise in farbigen Petri-Netzen Variablen an den Kanten stehen, die zum Schalten mit Farben bzw. Marken belegt werden, ist in diesen Fällen die Berechnung symbolischer Invarianten möglich und sinnvoll. Als symbolische Invariante sei hier eine Funktion über den Stellen bezeichnet, die sich als gewichtete Summe von Farbfunktionen über Stellen ausdrücken läßt. Eine solche symbolische Invariante kann für eine Familie von expliziten Invarianten stehen. In den hier verwendeten Netzklasse treten keine Variablen mehr auf, da alle vorkommenden Farben bereits bekannt sind und explizit in den Ein- und Ausgangsausdrücken der Transitionsübergänge enthalten sind. Darum beschränken wir uns hier auf die Berechnung von „expliziten“ Invarianten, die als gewichtete Summe von Markenanzahlen in Stellen ausdrückbar sind und in der Literatur auch als *vector flow* bezeichnet werden [35]. Als Koeffizienten werden nur natürliche Zahlen zugelassen. Die dabei berechneten Invarianten enthalten keine speziellen Funktionen über den auftretenden Markenfarben. Dies vereinfacht ihr Verständnis und verhindert die sonst mögliche Erzeugung einer großen Menge von Invarianten mit unklarer Bedeutung.

Zur Berechnung der Invarianten eignet sich eine Modellbeschreibung mit einer Inzidenzmatrix besser als ein grafisches Modell. Die Inzidenzmatrix J beschreibt für die Stellen die beim Schalten der Übergänge verbrauchten und neu erzeugten Marken. In dem hier verwendeten Gesamtmodell ist das Schaltverhalten der Transitionen durch die Übergänge beschrieben, deren Ein- und Ausgangsschaltverhalten u^{in} und u^{out} auch innerhalb einer Transition sehr unterschiedlich sein kann. Darum muß in der Inzidenzmatrix eine Spalte für jeden Übergang jeder Transitionen enthalten sein. Die Menge aller auftretenden Paare aus Transition und Übergang wird als **Übergangselemente** UE bezeichnet:

$$UE = \{(t, u) \mid t \in T \wedge u \in U(t)\}$$

Für die in den Stellen vorkommenden Farben wird eine Entfaltung vorgenommen, so daß für jede Kombination von Stelle und Farbe eine Zeile in der Inzidenzmatrix vorgesehen ist. In einer Implementierung sollte die Größe der Matrix sinnvoll verkleinert werden, indem nur die tatsächlich in der Stelle auftretenden Farben verwendet werden. Diese sind aus den Übergängen der mit der Kante verbundenen Transitionen einfach zu bestimmen. Die Menge aller Paare aus Stelle und Farbe wird als **Stellenelemente** PE bezeichnet:

$$PE = \{(p, c) \mid p \in P \wedge c \in C(p)\}$$

Im weiteren sei vorausgesetzt, daß das Ein- und Ausgangsverhalten der Übergänge nicht von der aktuellen Markierung abhängt wie in der Definition zugelassen, da dies die Berechnung der Invarianten in dieser Form unmöglich macht. Damit bilden u^{in} und u^{out} jeweils direkt Ein- bzw. Ausgangsstellen ihrer Transition auf Multimengen des den Stellen entsprechenden Farbtyps ab. Die Inzidenzmatrix J enthält dann nur natürliche Zahlen und hat die Dimension $|PE| \times |UE|$ ($|A|$ sei die Mächtigkeit einer Menge A). J ist definiert durch

$$\forall (p, c) \in PE, (t, u) \in UE : J_{(p,c),(t,u)} = \begin{cases} u^{out}(p)(c) - u^{in}(p)(c) & \text{falls } p \in P^{in}(t) \cup P^{out}(t) \\ 0 & \text{sonst} \end{cases}$$

Jeder Vektor $\psi \in \mathbb{N}^{|PE|}$, der die Gleichung

$$\psi^T J = 0$$

erfüllt, ist eine Stellen-Invariante [62]. Im folgenden bezeichne Ψ die Menge aller Stellen-Invarianten eines Modells, wobei die triviale Invariante, deren Koeffizienten alle Null sind, außer acht gelassen wird.

Der **Träger** \mathcal{T}_ψ einer Invariante ψ ist die Menge der Stellenelemente, deren Koeffizient in ψ größer Null ist:

$$\mathcal{T}_\psi = \{(p, c) \in PE \mid \psi_{(p,c)} > 0\}$$

Als **Invariante mit minimalem Träger** (*minimal support invariant*) wird eine Invariante $\psi^* \in \Psi$ genau dann bezeichnet, wenn ihr Träger \mathcal{T}_{ψ^*} keinen Träger einer anderen Invariante enthält, also gilt:

$$\forall \psi \in \Psi \setminus \{\psi^*\} : \mathcal{T}_{\psi^*} \not\subseteq \mathcal{T}_\psi$$

Mit Ψ^* wird die Menge der Stellen-Invarianten mit minimalem Träger bezeichnet.

Die Inzidenzmatrix J gleicht der entsprechenden Matrix eines durch Entfaltung des speziellen Petri-Netzes gewonnenen farblosen Netzes. Es ist bekannt, daß sich jede Invariante eines farblosen Netzes als positive Linearkombination von Invarianten mit minimalem Träger darstellen läßt [68]. Die Menge Ψ^* enthält demnach den interessanten Teil der Invarianten, da sich alle anderen daraus ableiten lassen.

Zum Rechnen mit Invarianten und deren Interpretation werden auch in eine farblose Repräsentation entfaltete Markierungen benötigt. Die **Markierung** r eines speziellen Petri-Netzes ist eine Funktion, die jede Stelle des Modells auf eine Multimenge von Farben abbildet. Diese Multimenge darf nur Farben enthalten, die für die jeweilige Stelle zugelassen sind (elementare oder zusammengesetzte):

$$r : \begin{cases} PE \rightarrow \mathcal{M}_E \\ P^\Phi \rightarrow \mathcal{M}_\Phi \end{cases}$$

Die Menge aller beliebigen Markierungen wird als R bezeichnet. Die Multimenge, die in einer Markierung einer Stelle zugeordnet ist, wird die **Markierung** oder **Inhalt der Stelle** in dieser Markierung genannt. Ist eine Farbe in dieser Multimenge enthalten, bezeichnet man sie als in dieser Stelle befindliche **Marke**. Da es sich bei der Markierung um eine Multimenge handelt, kann dieselbe Farbe mehrfach in einer Stelle enthalten sein. Man spricht dann zwar von verschiedenen Marken, diese sind aber aus Modellsicht nicht voneinander unterscheidbar.

In der Definition der Netzklasse wurde bereits darauf hingewiesen, daß Anschlußstellen als Verweise auf die zugeordneten Sockelstellen aufgefaßt werden können und darum alle Attribute der zusammengehörenden Stellen jederzeit gleich sein müssen. Dies gilt auch für die Markierung:

$$\forall (p_i, p_k) \in AZ, \forall r \in R : r(p_i) = r(p_k)$$

Als zusammengehörig werden dabei alle zu einer Sockelstelle gehörenden Anschlußstellen sowie die Sockelstelle selbst bezeichnet. Wird die AZ -Relation reflexiv und symmetrisch erweitert und darüber die transitive Hülle gebildet, entsteht eine Äquivalenzrelation, die Klassen zusammengehöriger Stellen bildet. Wenn im folgenden die Markierung einer Stelle durch einen Übergang verändert wird, bezieht sich die Veränderung einer Stelle stets auch stillschweigend auf alle derselben Klasse zugehörigen Stellen.

Einer Markierung r wird der entfaltete Markierungsvektor $\bar{r} \in \mathbb{N}^{|PE|}$ zugeordnet:

$$\forall r \in R, \forall (p, c) \in PE : \bar{r}_{(p,c)} = r(p)(c)$$

Mit dem in [54] beschriebenen Algorithmus lassen sich alle Invarianten mit minimalem Träger eines farblosen Petri-Netzes berechnen. Nach der beschriebenen Entfaltung ist er auch auf die speziellen Petri-Netze anwendbar. Seine Funktionsweise soll hier nicht näher erläutert werden. Verschiedene derartige Algorithmen werden in [73] miteinander verglichen. Das Ergebnis des genannten Algorithmus ist die Menge der Invarianten mit minimalem Träger Ψ^* für ein spezielles Petri-Netz. Die einzelnen Invarianten $\psi^* \in \Psi^*$ müssen anschließend für das farbige Modell interpretiert werden. Für jede Invariante ψ gilt:

$$\forall r \in R : \psi^T \bar{r} = \sum_{(p,c) \in PE} \psi_{(p,c)} \bar{r}_{(p,c)} = \text{konstant} = \psi^T \bar{r}_0,$$

wobei r_0 die initiale Markierung ist, aus der der Wert der Invarianten berechnet werden kann. Diese Gleichung kann in der im Modell verwendeten Schreibweise dargestellt werden, da der Wert von $\bar{r}_{(p,c)}$ gleich $\#p(c)$ ist. Ausdrücke dieser Art sind für den Modellierer leicht verständlich, da sie in der aus dem Modell bekannten Weise darstellbar sind.

Die beschriebene Vorgehensweise ergibt für das in den vorherigen Kapiteln dargestellte Beispiel unter anderem folgende Invarianten mit minimalem Träger, die hier in Modell-Schreibweise als Gleichung dargestellt werden: Auf der zur Transition $M1$ der Primärseite gehörenden Unterseite (Bibliotheksmodul siehe Seite 55) gilt $\#PAusgefallen(e) + \#POK(e) = 1$. Demnach ist die Maschine immer in genau einem ihrer beiden Zustände Ausgefallen oder in Ordnung.

Auf der Unterseite des fahrerlosen Transportsystems, die durch die Transition FTS auf der Primärseite modelliert wird (Bibliotheksmodul siehe Seite 53), gilt: $\#BeiL(e) + \#BeiM1(e) + \#BeiM2(e) = 1$. Demzufolge befindet sich das FTS-Fahrzeug immer genau an einem der dafür vorgesehenen Plätze, dem Lager oder einer der beiden Maschinen.

Jedes Werkstück innerhalb des Bereichs der Bearbeitungsstationen befindet sich immer auf einem Werkstückträger, deren Anzahl sich nicht verändert. Diesem Sachverhalt entspricht ebenfalls eine Stellen-Invariante, die durch den Algorithmus berechnet wird, aber hier aufgrund ihres Umfangs nicht dargestellt ist. Die genannten Invarianten korrespondieren mit realen Systemeigenschaften, deren Vorhandensein im Modell so überprüft werden kann.

Zur übersichtlicheren Darstellung von Invarianten könnten diese entsprechend dem Modell-aufbau hierarchisch dargestellt werden, indem der Inhalt von Vertretertransitionen zusammenfassend für den Inhalt der in ihrer Unterseite befindlichen Stellen verwendet wird. Die Ausdrücke zur Beschreibung der Invarianten können weiter vereinfacht werden. Dazu kann anhand der Menge der in einer Stelle vorkommenden Farben untersucht werden, ob z.B. alle diese Farben mit demselben Faktor im Ausdruck vorkommen. In diesem Fall können die entsprechenden Teilausdrücke durch einen Ausdruck der Art **#Stelle** ersetzt werden. Dies ist auch für andere sinnvoll zusammenfaßbare Teilmengen der Farbmenge denkbar, z.B. für alle Marken mit demselben Produktnamen, so daß Platzhalter verwendet werden können.

Ebenso wie Stellen-Invarianten können Transitions-Invarianten eines Modells berechnet werden. Eine solche Invariante ist ein Vektor $\psi \in \mathbb{N}^{|UE|}$, der für jedes Übergangselement eine natürliche Zahl enthält, und für den gilt:

$$J\psi = 0$$

Eine T-Invariante hat folgende Bedeutung: werden aus einer Markierung heraus alle Übergangselemente so oft geschaltet, wie es ihr Koeffizient in der Invariante angibt, wird wieder die Ausgangsmarkierung erreicht. Das heißt aber nicht, daß diese Übergangselemente entsprechend oft schaltfähig sein müssen. Sind sie es, wird die Invariante als realisierbar bezeichnet [62]. Da beispielsweise die Schaltbarkeitsprädikate nicht in die Berechnung eingehen, können sie die Realisierbarkeit einer Invariante verhindern.

Gegenseitiger Ausschluß

Zwei Übergangselemente $(t_i, u_j), (t_k, u_l) \in UE$ befinden sich im gegenseitigen Ausschluß (*mutual exclusion*), wenn sie in keiner Markierung des Modells gemeinsam schaltfähig sind [16]. Man schreibt dann $(t_i, u_j)ME(t_k, u_l)$. Für zwei Übergangselemente, die einander ausschließende Aktivitäten der Realität modellieren, muß diese Relation gelten. Anhand des Modells kann so nachgewiesen werden, daß beispielsweise einander aus Sicherheitsgründen widersprechende Aktionen nie gemeinsam durchgeführt werden. Diese Eigenschaft ist außerdem wichtig für die Berechnung von möglichen Konflikten (s.u.).

Für eine vollständige Untersuchung wäre die Aufstellung des gesamten Erreichbarkeitsgraphen notwendig, was aber für die Bestimmung der strukturellen Eigenschaften vermieden werden soll. Darum werden verschiedene hinreichende Bedingungen untersucht, die mit Hilfe der Netzstruktur berechnet werden können, und die im Begriff des strukturellen gegenseitigen Ausschlusses (*structural mutual exclusion, SME* [16]) zusammengefaßt werden.

Im einfachsten Fall haben zwei Übergangselemente unterschiedliche Prioritäten und können darum selbst dann nicht gemeinsam schalten, wenn die sonstigen Voraussetzungen dafür erfüllt wären. Obwohl in der hier verwendeten Netzklasse den Übergangselementen keine

Prioritäten zugeordnet sind, haben zeitlose Übergänge letztlich dadurch Priorität über zeitbehaftete, daß ihr Schalten sofort stattfindet. Es gilt $(t_i, u_j)SME(t_k, u_l)$ wenn:

$$(t_i \in T^{imm} \wedge t_k \in T^{tim})$$

Aus den Stellen-Invarianten läßt sich für zwei Übergangselemente nachweisen, daß in keiner Markierung genug Marken auf den Eingangsstellen vorhanden sein können, so daß beide schaltfähig sind. Dann sind diese Übergangselemente ebenfalls im gegenseitigen Ausschluß. Diese Relation wird als *marking mutual exclusion (MME)* bezeichnet. Unter der Bedingung, daß

$$\exists \psi^* \in \Psi^* : \sum_{(p,c) \in PE} \psi_{(p,c)}^* \max(\overline{u_j^{in}}_{(p,c)}, \overline{u_l^{in}}_{(p,c)}) > (\psi^* \bar{r}_0)$$

erfüllt ist, gilt $(t_i, u_j)MME(t_k, u_l)$. Die Relation *MME* kann noch erweitert werden, indem alle Stellen mit einer nicht markierungsabhängigen Kapazitätsfunktion aus den natürlichen Zahlen hinzugezogen werden. Diese Stellenmenge werde hier mit P^* bezeichnet. Dann würde auch gelten $(t_i, u_j)MME(t_k, u_l)$, wenn:

$$\exists p \in P^* : \sum_{c \in C(p)} \max(\overline{u_j^{in}}_{(p,c)}, \overline{u_l^{in}}_{(p,c)}) > K(p)$$

Die beiden genannten Relationen sind hinreichende Bedingungen für den tatsächlichen gegenseitigen Ausschluß:

$$(t_i, u_j), (t_k, u_l) \in UE : (t_i, u_j)MME(t_k, u_l) \vee (t_i, u_j)SME(t_k, u_l) \rightarrow (t_i, u_j)ME(t_k, u_l)$$

Im Beispiel läßt sich aus der oben genannten Invariante für das fahrerlose Transportsystem $\#Beil(e) + \#BeiM1(e) + \#BeiM2(e) = 1$ schlußfolgern, daß sich die Transitionen **LM1** und **ML1** im gegenseitigen Ausschluß befinden. Damit ist nachgewiesen, daß das FTS-Fahrzeug entsprechend der Modellspezifikation nicht gleichzeitig zum Lager und zur Maschine 1 fahren kann.

Konflikte

Ein Konflikt besteht zwischen zwei Übergangselementen, wenn beide in einer Markierung schaltfähig sind, durch das Schalten des einen aber das andere Übergangselement deaktiviert wird, also in der Folgemarkierung nicht mehr schaltfähig ist. Tritt ein Konflikt zwischen verschiedenen möglichen Aktivitäten auf, muß demnach eine Entscheidung darüber gefällt werden, welche der Aktionen durchgeführt wird. Wie dies geschieht, ist durch die Definition der verwendeten Netzklasse und des entsprechenden Verhaltens eindeutig festgelegt. Trotzdem kann es interessant sein, die in einem Modell auftretenden Konflikte zu entdecken, um möglicherweise ihre Auflösung genauer spezifizieren zu können. Dabei sind insbesondere solche Konflikte interessant, die nicht schon innerhalb der Arbeitsplanmodelle sichtbar sind.

Im Gegensatz dazu können Konflikte auch erst durch die Kombination der Arbeitsplanmodelle im Gesamtmodell entstehen, so daß sie erst da erkennbar sind. Dies könnte zum Beispiel für die Transition **Roboter** auf der Primärseite der Fall sein. Die Übergangstabelle dieser Transition hat die folgende Form:

```
[in#LagerAB=A.Geliefert, in#PLade=WT.leer, out#PLade=A.Beladen; ... |
 in#LagerAB=B.Geliefert, in#PLade=WT.leer, out#PLade=B.Beladen; ... |
 in#PLade=A.Montiert, out#PLade=WT.leer, out#FertigLager=A.Fertig; ... |
 in#PLade=B.Montiert, out#PLade=WT.leer, out#FertigLager=B.Fertig; ... ]
```

Die Transition kann also vier verschiedene Aktivitäten durchführen, die möglicherweise miteinander in Konflikt stehen. Die strukturelle Analyse zeigt aber, daß sich alle Übergangselemente im gegenseitigen Ausschluß befinden, denn die Stelle **PLade**, die Eingangsstelle für alle u^{in} ist, hat eine Kapazität von Eins. Demnach sind alle Übergangselemente *marking mutual exclusive* nach der erweiterten Definition.

Für die miteinander in Konflikt stehenden Übergangselemente, die darüber hinaus auch zum selben Zeitpunkt schalten können, ist die Angabe des Schaltgewicht u^{weight} wichtig, das die jeweilige relative Wahrscheinlichkeit für das Schalten dieses Übergangs angibt. Allgemein können Übergangselemente miteinander in Konflikt stehen, wenn sie sich gemeinsame Eingangsmarken entziehen, Stellen mit endlicher Kapazität in beider Nachbereich haben, oder sie durch ihr Schalten den Wert einer der Schaltbedingungen Falsch machen. Befinden sie sich dagegen im gegenseitigen Ausschluß, ist kein Konflikt möglich, da sie ja nie gemeinsam schaltfähig sind. Es können Mengen von möglicherweise miteinander im Konflikt stehenden Übergangselementen gefunden werden, die als erweiterte Konfliktmengen (*extended conflict sets* [16, 25]) bezeichnet werden. Alle diese Bedingungen sind anhand der Netzstruktur untersuchbar, allerdings ist die Berechnung des Einflusses auf Schaltbedingungen im allgemeinen Fall sehr kompliziert.

Erreichbarkeit

Oft stellt sich in technischen Systemen die Frage, ob gewisse Systemzustände erreicht werden können oder nicht. Der Nachweis, daß ein Zustand tatsächlich erreicht wird, ist nur durch Untersuchung des Erreichbarkeitsgraphen möglich.

Gilt eine der vorher berechneten Markierungsinvarianten für eine zu prüfende Markierung r nicht, ist sie nicht aus der Anfangsmarkierung r_0 heraus erreichbar. Dies entspricht der Gültigkeit von

$$\exists \psi^* \in \Psi^* : \psi^T \bar{r} \neq \psi^T \bar{r}_0$$

Für manche verbotene Zustände kann dadurch die Nichterreichbarkeit mit Hilfe von Invarianten nachgewiesen werden, deren Berechnung nur die Untersuchung der Netzstruktur erfordert.

Weitere Eigenschaften

Eine wichtige Eigenschaft eines Modells ist seine Beschränktheit, denn es können natürlich nur endliche Erreichbarkeitsgraphen aufgestellt und untersucht werden. Ein Modell ist beschränkt, wenn jede Stelle entweder mit allen ihren möglichen Farben in Stellen-Invarianten enthalten ist oder eine maximale Kapazität angegeben wurde. Alle anderen Stellen sind strukturell unbeschränkt und können demnach möglicherweise unendlich viele Markierungen

enthalten. Dies kann ein Hinweis auf einen Modellfehler sein. Insbesondere für die Produktstellen des Modells ist davon auszugehen, daß jeder potentielle Ort, an dem sich Werkstücke aufhalten können, in seiner Kapazität beschränkt ist. Aus den Stellen-Invarianten und den Kapazitäten der Stellen kann eine obere Schranke für die maximale Anzahl der Zustände berechnet werden, um eine Aufwandsabschätzung für spätere Analysevorgänge durchführen zu können. Eine entsprechende Methode ist in [51] für farblose Modelle beschrieben.

Zur Ableitung vieler einfacher Informationen aus dem Modell, die zur Kontrolle des Modelliervorgangs dienen können, sind ebenfalls nur Strukturinformationen notwendig. Zum Beispiel können für die einzelnen bearbeiteten Produkte die Erzeugnisstrukturbäume und Materialverwendungsnachweise, die Ein- und Ausgänge, die Bearbeitungs- und Montagestellen sowie Transportvorgänge und ähnliches aus dem Modell gewonnen werden. Außerdem können alle in einer Stelle vorkommenden Markenarten und die von einer Transition durchgeführten Aktivitäten (Übergänge) angegeben werden. Diese Informationen sind sehr einfach aus dem Modell zu gewinnen und erscheinen möglicherweise trivial, sind aber für die Überprüfung eines großen Modells hilfreich.

Eine wichtige Eigenschaft eines Fertigungssystems ist seine Verklemmungsfreiheit bzw. Lebendigkeit. Diese Eigenschaft sichert, daß das einmal gestartete System nicht in einen Zustand kommt, in dem es völlig blockiert ist. Ein spezielles Petri-Netz-Modell ist lebendig, wenn es in jedem aus dem initialen Zustand erreichbaren Zustand mindestens ein schaltfähiges Übergangselement gibt. Ein solches Modell ist offensichtlich verklemmungsfrei. Es gibt jedoch auch weitergehende Definitionen der Lebendigkeit, in der z.B. für jede erreichbare Markierung gefordert wird, daß jede Transition wieder aktivierbar ist.

Für einfache Netze kann die Lebendigkeit anhand von Invarianten geprüft werden. Beispielsweise ist jedes endliche, lebendige und beschränkte Stellen-Transitions-Netz von Transitions-Invarianten überdeckt [62]. Ist in diesem Fall also eine in keiner T-Invariante enthaltene Transition zu finden, kann ein endliches Modell nicht lebendig und beschränkt sein. Der Nachweis der Lebendigkeit eines Modells ist in allgemeinen Fällen nur anhand des Erreichbarkeitsgraphen überprüfbar. Enthält der Erreichbarkeitsgraph keine absorbierenden Zustände (s.u.), ist es lebendig. Dies ist ein Beispiel für eine qualitative Eigenschaft eines Modells, die nicht ausschließlich aus der Struktur ableitbar ist.

4.3 Quantitative Eigenschaften

Dieser Abschnitt beschäftigt sich mit der Gewinnung von Leistungs- und Zuverlässigkeitsaussagen über ein modelliertes Fertigungssystem. Es wird gezeigt, welche Möglichkeiten die spezielle Petri-Netz-Klasse dafür bereithält. Dazu wird zunächst das dynamische Verhalten eines speziellen Petri-Netzes dargestellt, welches für Simulation und Analyse gleichermaßen benötigt wird. Die für die Durchführung von numerischen Analysen und der Simulation notwendigen Techniken für die spezielle Netzklasse werden im Anschluß dargestellt und die Vor- und Nachteile beider Verfahren aufgezeigt.

Zur quantitativen Analysierbarkeit ist die Verwendung zeitbehafteter Vorgänge im Modell notwendig. Dabei wird auf Erweiterungen der stochastischen Petri-Netze aufgebaut [32], die vor allem die Abbildung von häufig in Fertigungssystemen auftretenden nicht-exponentiellen

Schaltzeiten ermöglichen. Die direkte numerische Analyse eines Modells hat gegenüber der Simulation den Vorteil, daß die Ergebnisse ohne Genauigkeitsabschätzung verwendet werden können und die Lösung oft schneller berechnet wird. Andererseits gibt es Modelle, die nicht analytisch untersucht werden können und darum eine Simulation erfordern. Dies ist der Fall, wenn der Erreichbarkeitsgraph zu groß ist oder wenn z.B. mehrere nichtexponentielle Übergänge in einem Zustand schaltfähig sind. Aus diesem Grund werden beide Untersuchungsmethoden dem Modellierer zur Verfügung gestellt.

In manchen Methoden und Werkzeugen zur Untersuchung von farbigen Petri-Netz-Modellen wird zunächst das Modell entfaltet, also auf ein farbloses Netz mit gleicher Funktionalität abgebildet. Dies hat den Vorteil, daß existierende Werkzeuge für die Untersuchung der farblosen Modelle eingesetzt werden können. Neben dem dabei auftretenden Mehraufwand der Umwandlung in ein anderes Modell ist dabei aber die Interpretation der aus dem farblosen Netz gewonnenen Ergebnisse problematisch. Darum wird in dem vorgeschlagenen Verfahren der Erreichbarkeitsgraph direkt aus dem farbigen Modell bestimmt, obwohl die Umwandlung in ein farbloses Modell prinzipiell möglich ist [13].

4.3.1 Untersuchung des Netzverhaltens

In diesem Abschnitt wird das Verhalten eines speziellen Petri-Netzes erklärt. Dabei muß vor allem geklärt werden, welche Zustände und Zustandsübergänge auftreten können.

Ein wichtiger Grund für die Verwendung farbiger Petri-Netze zur Modellierung besteht darin, daß sich mit ihnen Systeme mit Symmetrien gut beschreiben lassen. Das kann z.B. der Fall sein, wenn mehrere identische Prozesse ablaufen, die die gleichen Aufgaben erfüllen. Ein farbiges Petri-Netz eines solchen Systems kann dann als gefaltetes farbloses Netz interpretiert werden. Dann beschreibt eine Transition im farbigen Petri-Netz die Faltung bzw. Überlagerung verschiedener Transitionen aus äquivalenten farblosen Netzen. Alle diese Schaltmöglichkeiten einer Transition des farbigen Modells können deshalb gleichzeitig und konkurrierend miteinander schaltfähig sein. Diese Schaltsemantik wird hier als *infinite server* bezeichnet, da sie der entsprechenden Schaltsemantik für farblose Netze [3] ähnelt.

Im Gegensatz dazu modellieren zeitbehaftete Transitionen in den hier verwendeten Modellen von Fertigungssystemen meist Maschinen, Transporteinrichtungen oder andere Ressourcen. Diese können normalerweise nur eine Aktivität auf einmal durchführen. Die verschiedenen Schaltmöglichkeiten einer Transition beschreiben hier also die unterschiedlichen Aktivitäten, die die Maschine ausführen kann. Damit wie in der Realität nur eine solche Schaltmöglichkeit gleichzeitig aktiviert sein kann, wird eine als *exclusive server* bezeichnete Schaltsemantik für zeitbehaftete Transitionen eingeführt. Wird eine Transition dieser Art schaltfähig, muß eine Vorauswahl (*preselection*) zwischen den theoretisch schaltfähigen Schaltmöglichkeiten getroffen werden, die bis zur nächsten Deaktivierung der Transition bestehen bleibt. Dieses Verhalten wird in [3] für farblose Petri-Netze unter der Bezeichnung *local preselection* mit *persistent choice variables* beschrieben. Im Gegensatz zu farblosen Netzen, in denen die Spezifikation eines solchen Verhaltens schwer ist [3], kann in der hier vorgeschlagenen Netzklasse die Menge der Transitionen, zwischen denen eine Vorauswahl zu erfolgen hat (*preselection set*), leicht bestimmt werden. Diese Transitionen entsprechen den verschiedenen Schaltmöglichkeiten (Übergängen) der zeitbehafteten Transitionen, deren Schaltseman-

tik als *exclusive server* angegeben wurde. Für jede dieser Transitionen muß die getroffene Vorauswahl in der Zustandsbeschreibung gespeichert werden. Dies geschieht in sogenannten Vorauswahl-Variablen (*choice variables* oder *preselection variables*).

Zeitlose Transitionen modellieren keine realen Aktivitäten, sondern z.B. eine probabilistische Auswahl zwischen mehreren Möglichkeiten. Darum ist für sie nur die „normale“ Schaltsemantik *infinite server* sinnvoll und erlaubt. Für zeitbehaftete Transitionen sind in der verwendeten Netzart beide Schaltsemantiken zugelassen. Die Spezifikation erfolgt mit der Funktion *SS* (siehe Definition der Netzklasse).

Zustand eines speziellen Petri-Netzes

Aus diesen Überlegungen heraus ist klar, daß der **Zustand** z_i eines speziellen Petri-Netzes neben der aktuellen Markierung r_i zusätzlich durch eine Funktion PV_i beschrieben wird, die für jede nur exklusiv schaltfähige Transition t den Wert der entsprechenden Vorauswahl-Variable $PV_i(t)$, also den in dem Zustand aktivierten Übergang, angibt:

$$z_i = (r_i, PV_i)$$

Die Funktion PV bildet eine Transition t auf den aktivierten Übergang aus $U(t)$ ab, falls eine Vorauswahl getroffen wurde und der entsprechende Übergang u schaltfähig ist (der Begriff der Schaltfähigkeit wird weiter unten erklärt):

$$\forall t \in T^{tim}, SS(t) = exclusive : PV(t) = \begin{cases} u \in U(t) & \text{falls } u \text{ aktiviert} \\ 0 & \text{falls kein Übergang aktiv ist} \end{cases}$$

Wenn nicht anders angegeben, ist mit einem speziellen Petri-Netz in einer bestimmten Markierung der Zustand gemeint, in dem keine Vorauswahl getroffen wurde. Der Begriff der Markierung wurde bereits auf Seite 70 erläutert.

Der **initiale Zustand** $z_0 = (r_0, PV_0)$ des Petri-Netzes definiert den Zustand des Modells, bei dem eine Untersuchung beginnt. In ihm ist noch keine Vorauswahl getroffen, so daß gilt

$$\forall t \in T^{tim}, SS(t) = exclusive : PV_0(t) = 0.$$

Die **initiale Markierung** r_0 wird dagegen aus der Initialisierungsfunktion I gewonnen:

$$\forall p \in P : r_0(p) = I(p)$$

Ein Zustand $z_i = (r_i, PV_i)$ bzw. eine Markierung r_i heißen **zulässig**, wenn in ihnen „keine Regeln verletzt“ werden, d.h. die Kapazität K aller Stellen $p \in P$ durch ihre Markierung $r_i(p)$ nicht überschritten wird:

$$\forall p \in P : K(p, r_i) = Wahr$$

Die **Menge der zulässigen Zustände** wird mit Z bezeichnet. Der initiale Zustand muß natürlich zulässig sein:

$$z_0 \in Z$$

Zulässige Markierungen entsprechen realen Systemzuständen, die aufgrund technischer Begrenzungen denkbar sind.

Zustandsübergänge

Nach der Definition des Zustandsbegriffes wird nun das Verhalten des Modells mit den möglichen Übergängen zwischen den Zuständen beschrieben. Zustandsübergänge werden durch die aktiven Modellkomponenten, die Transitionen und ihre Übergänge, ausgelöst. In der Literatur werden verschiedene Schaltmechanismen für Transitionen beschrieben. Von diesen wird hier der wohl gebräuchlichste verwendet, in dem das Schalten einer Transition (bzw. eines Übergangs) atomar ist. Das bedeutet, daß das Abziehen der Marken aus den Eingangsstellen und das Erzeugen neuer Marken in den Ausgangsstellen ohne Unterbrechung abläuft.

Um die Dauer von Aktivitäten spezifizieren zu können, werden den Übergängen Zufallsvariablen zugeordnet, die die Verteilung der Schaltzeit $u^{delay} \in \mathcal{F}$ beschreiben. Solange in jedem Zustand höchstens ein zeitbehafteter Übergang schaltfähig ist, dessen Dichtefunktion der Schaltzeitverteilung Sprünge enthält bzw. nicht stetig differenzierbar ist (dies ist Voraussetzung für die Anwendbarkeit der in Kapitel 4.3.2 vorgestellten numerischen Analysetechnik), ist die theoretische Wahrscheinlichkeit für das gleichzeitige Schalten von zeitbehafteten Übergängen gleich Null [32]. Dies liegt daran, daß ein zeitkontinuierliches Modell verwendet wird. Der zugrundeliegende Prozeß wird als *continuous-time discrete-state stochastic point process* bezeichnet [20]. Im Falle der Simulation kann es dagegen sehr wohl passieren, daß zwei zeitbehaftete Übergänge zum selben Zeitpunkt versuchen zu schalten. Für zeitlose Übergänge, die gemeinsam schaltfähig sind, ist dies dagegen immer der Fall. Im Gegensatz zu anderen Vorschlägen aus der Literatur, in denen mehrere gleichzeitig schaltfähiger Übergänge gemeinsam schalten, werden alle zum selben Zeitpunkt schaltenden Übergänge hier sequentiell nacheinander ausgeführt (*interleaving semantic*). Um entscheiden zu können, welcher der Übergänge zuerst auszuführen ist, wird das Schaltgewicht u^{weight} herangezogen. Die Wahrscheinlichkeit, daß ein Schritt zuerst ausgeführt wird, ist gleich dem Schaltgewicht dieses Übergangs geteilt durch die Summe aller Schaltgewichte der Übergänge, die in diesem Zeitpunkt schalten.

Nachdem eine Transition schaltfähig geworden ist, wird ein zufälliger Wert aus der Zufallsvariablen der Schaltzeit $u^{delay} \in \mathcal{F}$ gezogen. Dieser Wert entspricht der Zeit, die vergeht, bevor die Transition schaltet. Natürlich kann während dieser Zeit eine andere Aktivität stattfinden, so daß der Übergang nicht mehr schaltfähig ist, sondern unterbrochen wird. Für die Auswahl der nächsten zu schaltenden Transition und das Gedächtnis der Schaltzeiten sind verschiedene Semantiken untersucht worden [3].

Die *execution policy* bestimmt, welcher schaltfähige Zustandsübergang als nächster geschaltet wird. Es liegt nahe, den Übergang mit der kleinsten Schaltzeit auszuwählen (*race policy*). Im Fall einer aktivierten Transition mit der Schaltsemantik *exclusive server* wird dagegen eine von den zugeordneten Auswahlwahrscheinlichkeiten abhängige Vorauswahl getroffen (s.o.). Die für die speziellen Petri-Netze verwendete *execution policy* kann demnach als *race* im Normalfall und *local preselection* mit *persistent choice variables* [3] für Transitionen mit der Schaltsemantik *ES* charakterisiert werden. Durch die unten beschriebene Vorgehensweise beim Bestimmen schaltfähiger Schritte werden Zwischenzustände eingeführt, in denen die Vorauswahl durchgeführt wird.

Als zweites muß die *memory policy* spezifiziert werden. Sie bestimmt, auf welche Art die den Zustandsübergängen zugeordneten Schaltzeiten durch einen Zustandswechsel beeinflusst wer-

den. Aktivitäten sollen unterbrechbar (d.h. deaktivierbar) sein, aber nicht von anderen Zustandswechseln beeinflusst werden. Darum wird die *resampling policy* verwendet. Die bereits zurückgelegte Schaltzeit von Übergängen, die durch das Schalten eines anderen Übergangs deaktiviert werden, ist verloren. Sobald der Übergang wieder aktiviert wird, muß eine neue Schaltzeit bestimmt werden. Durch die Gedächtnislosigkeit der Exponentialverteilung ist die Angabe der *memory policy* nur für Übergänge mit nichtexponentieller Schaltzeitverteilung interessant [32], da sie nur hier einen Einfluß auf das Verhalten hat.

Ein **Schritt** (Zustandsübergang in einem speziellen Petri-Netz) ist ein geordnetes Paar aus einer Transition t und einem Übergang $u \in U(t)$. Ein solcher Schritt $\theta = (t, u)$ heißt **zulässig im Zustand** $z_i = (r_i, PV_i)$, $z_i \in Z$ genau dann wenn die folgenden Bedingungen erfüllt sind:

- Die Schaltbedingung $G(t)$ der Transition t (ein möglicherweise markierungsabhängiger boolescher Ausdruck) muß bei der Auswertung in der Markierung r_i wahr sein:

$$G(t, r_i) = \text{Wahr}$$

- Die Schaltbedingung u^{guard} des Übergangs u , ebenfalls ein möglicherweise markierungsabhängiger boolescher Ausdruck, muß bei der Auswertung in der Markierung r_i wahr sein:

$$u^{guard}(r_i) = \text{Wahr}$$

- Auf den Eingangsstellen von t sind genug Marken für das Schalten von u vorhanden:

$$\forall p \in P^{in}(t) : r_i(p) \geq u^{in}(p, r_i)$$

- Durch das Schalten von t wird für keine der beeinflussten Stellen die Kapazität überschritten. Dabei sei $r'_i \in R$, die „Folgemarkierung“ des Schrittes, wie folgt definiert:

$$\forall p \in P : r'_i(p) = r_i(p) - u^{in}(p, r_i) + u^{out}(p, r_i)$$

dann muß r'_i eine zulässige Markierung sein (siehe oben).

Die Menge der im Zustand z zulässigen Schritte wird mit Θ_z bezeichnet.

Ein im Zustand z zulässiger Schritt $\theta \in \Theta_z$ wird als **schaltfähig im Zustand** z bezeichnet, wenn er in der Menge der im Zustand z schaltfähigen Schritte Θ_z^* (siehe unten) enthalten ist. Die Menge der zulässigen Zustände Z wird wie folgt eindeutig in vier paarweise disjunkte Mengen eingeteilt, für die jeweils die Menge der in ihnen schaltfähigen Schritte definiert wird:

- Die Menge der **verschwindenden Zustände** Z^{imm} ist die Teilmenge der zulässigen Zustände, in denen mindestens ein zulässiger Schritt mit einer zeitlosen Transition existiert:

$$Z^{imm} = \{z \in Z \mid \exists (t, u) \in \Theta_z \wedge t \in T^{imm}\}$$

Für einen verschwindenden Zustand z ist die Menge der schaltfähigen Schritte Θ_z^* wie folgt definiert:

$$\forall z \in Z^{imm} : \Theta_z^* = \{(t, u) \in \Theta_z \mid t \in T^{imm}\}$$

- Die Menge der **Zwischenzustände** Z^{int} ist die Teilmenge der zulässigen Zustände, die keine verschwindenden Zustände sind, und in denen es mindestens einen zulässigen Schritt einer zeitbehafteten Transition mit *exclusive server* Schaltsemantik gibt, deren Vorauswahl-Funktion 0 ist:

$$Z^{int} = \{z \in Z \mid z \notin Z^{imm} \wedge \exists(t, u) \in \Theta_z \wedge t \in T^{tim} \wedge SS(t) = exclusive \wedge PV(t) = 0\}$$

Für einen Zwischenzustand z ist die Menge der schaltfähigen Schritte Θ_z^* wie folgt definiert:

$$\forall z \in Z^{int} : \Theta_z^* = \{(t, u) \in \Theta_z \mid t \in T^{tim} \wedge SS(t) = exclusive \wedge PV(t) = 0\}$$

- Die Menge der **andauernden Zustände** Z^{tim} ist die Teilmenge der zulässigen Zustände, die weder verschwindende noch Zwischenzustände sind, und in denen es mindestens einen zulässigen Schritt gibt:

$$Z^{tim} = \{z \in Z \mid z \notin Z^{imm} \cup Z^{int} \wedge \exists(t, u) \in \Theta_z\}$$

Dies ist gleichbedeutend damit, daß es mindestens einen schaltfähigen und zeitbehafteten Übergang in diesem Zustand gibt. Für einen andauernden Zustand z ist die Menge der schaltfähigen Schritte Θ_z^* wie folgt definiert:

$$\forall z \in Z^{tim} : \Theta_z^* = \{(t, u) \in \Theta_z \mid SS(t) = exclusive \Rightarrow PV(t) = u\}$$

Demnach ist für Transitionen mit der Schaltsemantik *exclusive server* nur der gerade aktivierte Übergang schaltfähig. Im Gegensatz dazu sind für schaltfähige Transitionen mit *infinite server* Schaltsemantik alle zulässigen Übergänge schaltfähig. Dabei ist zu beachten, daß jeder Übergang dabei auch mehrfach mit sich selbst konkurrierend schaltfähig sein kann. Die Schaltsemantik *infinite server* bedeutet prinzipiell, daß für jeden wartenden Kunden ein Bedienprozeß vorhanden ist. Wenn man dies auf die speziellen Petri-Netze überträgt, muß für jedes Mal, wenn ein Übergang zusätzlich schaltfähig wird, eine weitere Schaltzeit aus deren Verteilungsfunktion bestimmt werden. Aufgrund der Gedächtnislosigkeit der Exponentialverteilung kann dieses Verhalten für Transitionen mit exponentialverteilter Schaltzeit vereinfacht nachgebildet werden. Dazu wird bestimmt, wie oft ein Übergang aktiviert ist bzw. wie viele Male er sofort nacheinander schalten könnte. Die mittlere Schaltrate des Übergangs, die gleichzeitig Parameter der Exponentialverteilung ist, muß dann lediglich mit diesem Grad der Schaltfähigkeit multipliziert werden. Transitionen mit der Schaltsemantik *infinite server* dürfen bei numerischer Analyse keine nichtexponentielle Schaltzeitverteilung haben.

- Die Menge der **absorbierenden Zustände** Z^{abs} ist die Teilmenge der zulässigen Zustände, in denen kein zulässiger Schritt existiert:

$$Z^{abs} = \{z \in Z \mid \Theta_z = \emptyset\}$$

Da es in diesen Zuständen keine zulässigen Schritte gibt, können auch keine schaltfähigen Schritte existieren:

$$\forall z \in Z^{abs} : \Theta_z^* = \emptyset$$

Durch diese Unterteilung werden alle zulässigen Zustände erfaßt. Wird während der Untersuchung des Modells ein absorbierender (toter) Zustand erreicht, kann über eine Untersuchung dieses Zustands eine möglicherweise fehlerhafte Funktionsweise des Modells gefunden werden. Modelle mit absorbierenden Zuständen sind nur für die transiente Untersuchung quantitativer Größen interessant.

Ein Übergang heißt schaltfähig in einem Zustand genau dann, wenn es einen schaltfähigen Schritt in diesem Zustand gibt, der diesen Übergang enthält. Eine Transition wird in einem Zustand als schaltfähig bezeichnet, wenn mindestens einer ihrer Übergänge in diesem Zustand schaltfähig ist.

Durch die obige Festlegung der Schaltfähigkeit eines Schrittes ist sichergestellt, daß bei der Untersuchung des Modellverhaltens (z.B. der Berechnung des Erreichbarkeitsgraphs) folgende Reihenfolge beachtet wird:

- In einem Zustand werden zuerst immer alle zeitlosen Transitionen geschaltet, bis keine solche mehr schaltfähig ist. Zeitlose Transitionen modellieren Entscheidungen, die aus Modellsicht in vernachlässigbar kurzer Zeit stattfinden. Aus diesem Grund ist ihr Schalten gegenüber anderen Transitionen priorisiert.
- Wenn in einem Zustand keine zeitlosen Transitionen mehr schaltfähig sind, muß für alle *exclusive server* Transitionen untersucht werden, ob eine Auswahl eines ihrer schaltfähigen Übergänge getroffen werden muß. Dies ist der Fall, wenn sie im aktuellen Zustand neu schaltfähig geworden ist oder wenn sie gerade selbst geschaltet hat. Der Wert von $PV(t)$ bleibt gleich einem Übergang u , solange dieser zulässig ist. Damit wird modelliert, daß nach einer einmal getroffenen Entscheidung über eine auszuführende Aktivität diese bis zu Ende ausgeführt wird, wenn ihre Ausführung nicht schon vorher unmöglich gemacht wird. Dieses Verhalten entspricht *persistent preselection variables* und *enabling memory policy* nach [3]. Die Auswahl der zu schaltenden Übergänge muß vor dem eigentlichen Schalten dieser erfolgen, andernfalls könnte das Verhalten undefiniert sein.
- Erst wenn die beiden genannten Schrittarten nicht mehr ausgeführt werden müssen, können die Zeit verbrauchenden Übergänge geschaltet werden. Diese haben deshalb die geringste Priorität.

Schaltfähige Schritte beschreiben unteilbare Zustandsübergänge. Im folgenden wird definiert, wie ein solcher Schritt ausgeführt wird. Dabei wird wiederum nach drei Zustandsarten unterschieden, und es wird jeweils angegeben, wie sich der Folgezustand aus dem alten Zustand sowie dem schaltenden Schritt ergibt. Die Menge der absorbierenden Zustände wird hier außer acht gelassen, da in ihnen ja keine Schritte schaltfähig sind. Generell gilt, daß die Werte der markierungsabhängigen Ausdrücke vor dem Schalten eines Übergangs ausgewertet werden und dann für den gesamten Schaltvorgang beibehalten werden, auch wenn sich ihr Wert durch das Schalten selbst verändert.

- Das Schalten eines schaltfähigen Schrittes (t, u) in einem verschwindenden Zustand $z \in Z^{imm}$, $z = (r, PV)$ ist ein zeitloser Übergang zu einem Folgezustand $z' = (r', PV')$.

Falls mehrere Schritte schaltfähig sind, kann jeder nur mit einer gewissen Wahrscheinlichkeit ausgeführt werden. Dazu wird das Gewicht des schaltenden Übergangs mit der Summe der Gewichte aller schaltfähigen Schritte ins Verhältnis gesetzt:

$$P(\text{Übergang } (t, u) \text{ schaltet in } z) = \frac{u^{weight}}{\sum_{(t_i, u_k) \in \Theta_z} u_k^{weight}}$$

Dabei wird die Markierung r in r' verändert:

$$\forall p \in P : r'(p) = r(p) - u^{in}(p, r) + u^{out}(p, r)$$

In z' muß für alle *exclusive server* Transitionen die Vorauswahlfunktion PV auf Null zurückgesetzt werden, falls der aktivierte Übergang in z' nicht mehr zulässig ist. Demnach gilt:

$$\forall t \in T^{tim}, SS(t) = exclusive : PV'(t) = \begin{cases} 0 & \text{falls } (t, PV(t)) \notin \Theta_{z'} \\ PV(t) & \text{sonst} \end{cases}$$

- Das Schalten eines schaltfähigen Schrittes (t, u) in einem Zwischenzustand $z \in Z^{int}$, $z = (r, PV)$ bedeutet das Auswählen (Aktivieren) eines Übergangs einer *exclusive server* Transition. Es handelt sich dabei um einen zeitlosen Übergang zu einem Folgezustand $z' = (r, PV')$. Da das Schalten lediglich eine Auswahl eines aktivierten Übergangs beinhaltet, wird nur PV , nicht aber r durch das Schalten verändert. Falls mehrere zu einer Transition gehörende Schritte schaltfähig sind, kann jeder nur mit einer gewissen Wahrscheinlichkeit ausgeführt (aktiviert) werden. Dazu wird das Übergangsgewicht des schaltenden Übergangs mit der Summe der Übergangsgewichte aller schaltfähigen Schritte derselben Transition ins Verhältnis gesetzt:

$$P(\text{Übergang } (t, u) \text{ schaltet in } z) = \frac{u^{select}}{\sum_{(t, u_i) \in \Theta_z} u_i^{select}}$$

Dabei wird ausschließlich die Vorauswahlfunktion verändert:

$$PV'(t) = u$$

Bemerkung: Zwischenschritte können in beliebiger Reihenfolge ausgeführt werden, bis kein Zwischenzustand mehr vorliegt, da sich die Auswahl der PV der Transitionen nicht gegenseitig beeinflusst. Ist der Folgezustand kein Zwischenzustand, muß er ein andauernder Zustand sein.

- Das Schalten eines schaltfähigen Schrittes (t, u) in einem andauernden Zustand $z \in Z^{tim}$, $z = (r, PV)$ ist ein nach einer gewissen Zeit stattfindender Übergang zu einem Folgezustand $z' = (r', PV')$. Falls mehrere Schritte schaltfähig sind, wird entsprechend der Schaltsemantik *race* derjenige Übergang mit der kleinsten verbleibenden Schaltzeit ausgewählt. Wenn diese Zeit bei mehreren Schritten gleich ist, wird der zuerst zu schaltende Schritt wie bei den verschwindenden Zuständen anhand der Wahrscheinlichkeiten u^{weight} ausgewählt. Dies ist jedoch nur für die Simulation von Bedeutung; bei der

Analyse ist die Wahrscheinlichkeit für das gleichzeitige Schalten zweier zeitbehafteter Übergänge gleich Null. Durch das Schalten wird die Markierung verändert:

$$\forall p \in P : r'(p) = r(p) - u^{in}(p, r) + u^{out}(p, r)$$

In z' muß für alle *exclusive server* Transitionen die Vorauswahl-Funktion PV auf Null zurückgesetzt werden, falls der aktivierte Übergang in z' nicht mehr zulässig ist. Demnach gilt:

$$\forall t \in T^{tim}, SS(t) = exclusive : PV'(t) = \begin{cases} 0 & \text{falls } (t, PV(t)) \notin \Theta_{z'} \\ PV(t) & \text{sonst} \end{cases}$$

Erreichbarkeitsgraph

Ein Zustand z' heißt **direkt erreichbar** von einem Zustand z , wenn es einen in z schaltfähigen Schritt θ gibt, der z in z' überführt. Man schreibt dafür

$$z \xrightarrow{\theta} z'$$

Eine (endliche) **Schaltfolge** ist eine Folge von n Zuständen und $n - 1$ Schritten

$$z_1 \xrightarrow{\theta_1} z_2 \xrightarrow{\theta_2} \dots \xrightarrow{\theta_{n-1}} z_n,$$

wobei jeweils Zustand z_{i+1} direkt erreichbar ist aus Zustand z_i durch das Schalten des Schrittes θ_i . Dabei seien auch leere Schaltfolgen zugelassen; ein Zustand ist demnach immer aus sich selbst erreichbar. Ein Zustand z' heißt dann **erreichbar** von einem Zustand z , wenn es eine endliche Schaltfolge Θ gibt, die z in z' überführt. Ist dies der Fall, schreibt man auch

$$z \xrightarrow{\Theta} z'$$

Die **Erreichbarkeitsmenge** (*reachability set*) RS eines speziellen Petri-Netzes ist die Menge der Zustände, die alle aus dem initialen Zustand z_0 erreichbaren Zustände enthält:

$$RS = \{z \in Z \mid z_0 \xrightarrow{\Theta} z\}$$

Der **Erreichbarkeitsgraph** (*reachability graph*) RG eines speziellen Petri-Netzes ist ein gerichteter Graph, dessen Knotenmenge der Erreichbarkeitsmenge entspricht und dessen Kanten die schaltfähigen Übergänge beschreiben.

Um Werte für die Leistungsparameter zu bestimmen, werden Aufenthaltswahrscheinlichkeiten für die Modellzustände berechnet. Der Erreichbarkeitsgraph kann durch einen rekursiven Algorithmus ausgehend vom Anfangszustand bestimmt werden. In Zwischenzuständen und verschwindenden Zuständen verbringt der durch das beschriebene Modellverhalten gebildete stochastische Prozeß keine Zeit. Die Aufenthaltswahrscheinlichkeit dieser Zustände ist darum Null. Aus der Literatur sind verschiedene Verfahren bekannt, die diesen Umstand ausnutzen [12, 8, 51]. Dabei wird die numerische Analyse vereinfacht, indem die verschwindenden Zustände bereits bei der Erzeugung des Erreichbarkeitsgraphen eliminiert werden. Dieses

Verfahren kann für die hier verwendeten speziellen Petri-Netze auf die Zwischenzustände erweitert werden.

Die Grundidee ist, nach dem Schalten eines zeitbehafteten Übergangs alle folgenden Pfade zeitloser Schritte und Zwischenschritte zu nachfolgenden andauernden Zuständen rekursiv zu durchsuchen. Jedem Pfad kann dann eine zusammengefaßte Wahrscheinlichkeit zugeordnet werden, da sich bei Verzweigungen die Wahrscheinlichkeiten multiplizieren und bei Zusammenführung addieren. Falls der vorherige zeitbehaftete Schritt eine exponentielle Schaltzeitverteilung hatte, können die zeitlosen Pfade und der zeitbehaftete Schritt ersetzt werden, indem für jeden Pfad ein zeitbehafteter Übergang vom zeitbehafteten Zustand zum Endzustand des Pfades eingesetzt wird. Die Schaltraten der neuen Übergänge mit exponentieller Schaltzeit ergeben sich dabei aus dem Produkt der ursprünglichen Schaltrate und der jeweiligen Pfadwahrscheinlichkeit. Die alten Zustandsübergänge und zeitlosen Zustände können so eliminiert werden, während der Erreichbarkeitsgraph aufgebaut wird. Diese Methode wird als *elimination on the fly* bezeichnet [12], und führt in den meisten Fällen zu einer Effizienzsteigerung des Algorithmus.

Ist die Schaltzeitverteilung des zeitbehafteten Schrittes nicht rein exponentiell, müssen die zusammengefaßten Pfade mit ihren Wahrscheinlichkeiten aufbewahrt werden und können nicht mit dem zeitbehafteten Schritt zusammengefaßt werden.

Als **reduzierten Erreichbarkeitsgraph** eines Petri-Netzes bezeichnet man einen gerichteten Graphen, dessen Knoten der Menge der andauernden Zustände Z^{tim} des Modells entspricht. Er wird mit der eben beschriebenen Methode aus dem vollständigen Erreichbarkeitsgraphen gewonnen bzw. ohne Umweg direkt berechnet. Entsprechende Algorithmen sind in [51, 8] für farblose Modelle dargestellt. Sie lassen sich anhand der beschriebenen Eigenschaften der Zustände und Zustandsübergänge eines speziellen Petri-Netzes auf diese übertragen und werden hier nicht näher erläutert. Für jeden Zustandsübergang mit nichtexponentieller Schaltzeit müssen außerdem die zusammengefaßten Pfadwahrscheinlichkeiten zu folgenden andauernden Zuständen bekannt sein. Die Kanten des reduzierten Erreichbarkeitsgraphen beschreiben darum einen zeitbehafteten Zustandsübergang mit seiner Schaltzeitverteilung und möglicherweise nachfolgende Pfade von Übergängen, die keine Zeit verbrauchen.

Ein Beispiel

Zum besseren Verständnis des in diesem Abschnitt definierten Verhaltens folgt ein kleines Beispiel, das in Bild 4.1 gezeigt wird.

Der Einfachheit halber wurden nur Elementarstellen und entsprechend ununterscheidbare Marken verwendet. Bis auf die zeitbehaftete Transition T2, die zwei Übergänge besitzt, haben alle anderen Transitionen nur einen Übergang. Das Eingangs- und Ausgangsverhalten entspreche für alle Übergänge den aus der Grafik ersichtlichen Kanten, womit alle u^{in} und u^{out} festgelegt sind. Keine der Transitionen besitzt eine besondere Schaltbedingung u^{guard} . Die Auswahlwahrscheinlichkeit u^{select} ist nur für die Transition T2 mit der Schaltsemantik *exclusive server* interessant und sei für die beiden Übergänge $u_1^{select} = 2$ bzw. $u_2^{select} = 3$. Die Schaltzeitverteilung u^{delay} ist ebenfalls nur für die zeitbehaftete Transition T2 wichtig, wobei die tatsächlichen Werte von u_1^{delay} und u_2^{delay} hier keine Rolle spielen. Das Schaltgewicht u^{weight} ist wegen des zwischen ihnen auftretenden Konfliktes nur für die beiden zeitlosen

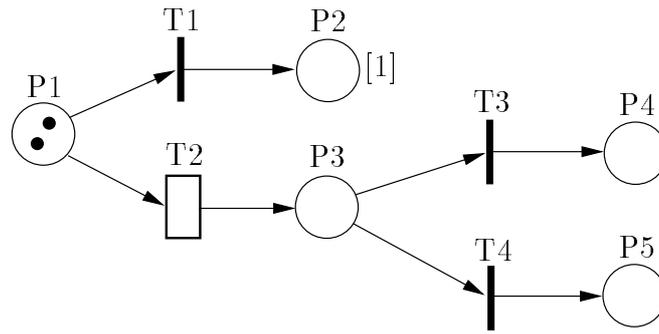


Abbildung 4.1: Beispiel-Modell zur Erklärung des Verhaltens

Transitionen **T3** und **T4** von Bedeutung und sei für **T3** gleich Eins, für **T4** gleich Zwei. Die Stelle **P2** hat die Kapazität Eins.

Der initiale Zustand wird durch Auswertung der Initialisierungsfunktion gewonnen und ist im Bild 4.1 dargestellt. Die Vorauswahlfunktion *PV* der Transition **T1** ist mit 0 initialisiert.

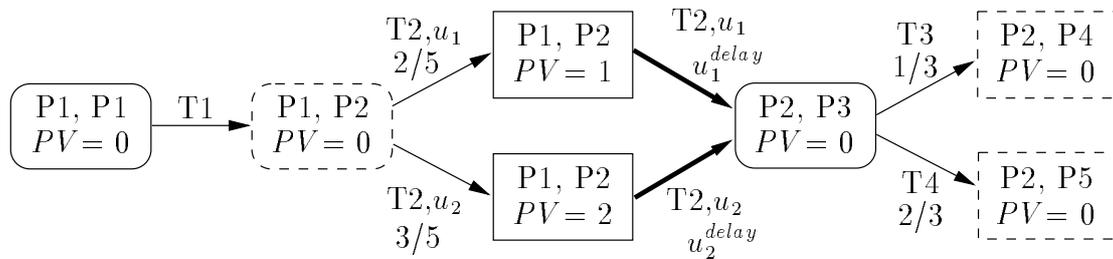


Abbildung 4.2: Erreichbarkeitsgraph für das Beispiel-Modell

Das Bild 4.2 zeigt den Erreichbarkeitsgraph des Modells. Zustände sind als Kästen dargestellt, in denen oben die Orte der beiden Marken und unten der Zustand der Vorauswahlvariablen der Transition **T2** eingetragen sind. Pfeile symbolisieren Zustandsübergänge, die mit den entsprechenden Transitionen und Übergängen beschriftet sind. Dabei bedeutet

- ein eckiger Kasten einen andauernden Zustand und ein gestrichelter eckiger Kasten einen absorbierenden Zustand,
- ein abgerundeter Kasten einen verschwindenden Zustand und ein ebensolcher gestrichelter Kasten einen Zwischenzustand,
- ein dünner Pfeil einen zeitlosen Schritt; dabei wird die Wahrscheinlichkeit dieses Schrittes angegeben, falls sie nicht gleich Eins ist,
- ein dicker Pfeil einen Schritt, der durch das Schalten einer zeitbehafteten Transitionen ausgelöst wird; zusätzlich ist die Schaltzeitverteilung angegeben.

Im initialen Zustand ist das Schalten des Übergangs von **T1** zulässig, es handelt sich also um einen verschwindenden Zustand. Demnach kann die zeitbehaftete Transition **T2** nicht schalten, obwohl ihre beiden Übergänge zulässig sind. Nachdem **T1** geschaltet hat, befindet sich eine der Marken in der Stelle **P2**. Damit ist die maximale Kapazität dieser Stelle erreicht,

und T1 kann nicht noch einmal schalten, da sonst ein unzulässiger Zustand erreicht werden würde.

Nun sind nur die zwei Übergänge von T2 zulässig, und die Vorauswahlfunktion von T2 ist Null. Der Zustand ist darum ein Zwischenzustand, in dem die Auswahl zwischen den zulässigen Übergängen von T2 getroffen wird. Dies geschieht mit den Auswahlwahrscheinlichkeiten, die die relative Wahrscheinlichkeit jedes Übergangs angeben. Im Ergebnis wird u_1 mit der Wahrscheinlichkeit $\frac{2}{2+3}$ und u_2 mit der Wahrscheinlichkeit $\frac{3}{2+3}$ ausgewählt. Dadurch werden zwei andauernde Zustände erreicht, in denen jeweils einer der Übergänge von T2 schaltfähig ist. Nach der entsprechenden Schaltzeit wird zum nächsten, gemeinsamen Folgezustand geschaltet. Dieser ist ein verschwindender Zustand, da die Übergänge der zeitlosen Transitionen T3 und T4 zulässig sind. Die Transitionen befinden sich miteinander im Konflikt, und die Wahrscheinlichkeit des Schaltens einer der beiden Transitionen muß über die Schaltwahrscheinlichkeiten u^{weight} berechnet werden. T3 schaltet mit der Wahrscheinlichkeit $\frac{1}{1+2}$ und T4 mit der Wahrscheinlichkeit $\frac{2}{1+2}$. Durch das Schalten jeder Transition wird ein absorbierender Zustand erreicht, in dem keine Übergänge mehr zulässig sind. Der Erreichbarkeitsgraph ist damit vollständig.

4.3.2 Numerische Analyse

Dieser Abschnitt beschreibt, wie aus einem speziellen Petri-Netz-Modell Leistungsparameter mit numerischen Verfahren gewonnen werden können. Dabei wird auf eine in [32] entwickelte numerische Analysetechnik für erweiterte deterministische und stochastische Petri-Netze (*eDSPNs*) bzw. semi-regenerative stochastische Petri-Netze (*SR-SPNs*) zurückgegriffen. Auf der Ebene der reduzierten Erreichbarkeitsgraphen kann diese Methode auch für die hier eingesetzten speziellen Petri-Netze verwendet werden, da der sich aus dem Modell ergebende stochastische Prozeß derselben Klasse angehört.

Bedingung für die numerische Analysierbarkeit von zeitbehafteten Petri-Netzen ist die Endlichkeit ihres Erreichbarkeitsgraphen. Für die Berechnung des eingeschwungenen Zustands wird vorausgesetzt, daß der Erreichbarkeitsgraph keine absorbierenden Zustände enthält. Wäre dies nicht der Fall, bestünde der eingeschwungene Zustand lediglich aus einer Wahrscheinlichkeitsverteilung zwischen diesen absorbierenden Zuständen. Die den Übergängen zugeordneten Schaltzeitverteilungen u^{delay} haben großen Einfluß auf die Analysierbarkeit eines Modells sowie die Komplexität der Lösungsberechnung. Die Menge der zeitbehafteten Übergänge kann unterteilt werden in solche mit rein exponentieller Schaltzeitverteilung in jeder Markierung U^{exp} und mit anderer Schaltzeitverteilung U^{gen} . Die für die Analyse verwendete Modellklasse erlaubt in jedem Zustand maximal einen schaltfähigen Übergang mit nichtexponentieller Schaltzeit, hier also aus U^{gen} .

Die Menge der nichtexponentiellen Schaltzeitverteilungen (*probability distribution function* oder *PDF*) aus \mathcal{F} ist auf die Menge der sogenannten expolynomialen Verteilungen [32] eingeschränkt. Funktionen dieser Art sind stückweise aus Kombinationen von Polynomen und Exponentialfunktionen zusammengesetzt. Die Wahrscheinlichkeitsdichtefunktion (*probability mass function, pmf* bzw. *probability density function, pdf*) kann durch die Ableitung dieser Funktion berechnet werden, die wieder eine expolynomialen Funktion ist, zusätzlich aber Impulse enthalten kann. Sie ist demnach eine Mischform aus kontinuierlichen (*pdf*) und

diskreten (*pmf*) Teilen, die hier aber vereinfachend als Wahrscheinlichkeitsdichtefunktion *pmf* bezeichnet werden soll. Jede Exponentialfunktion $F(t) \in \mathcal{F}$ läßt sich im Intervall $t_i \leq t < t_{i+1}$ ausdrücken durch

$$F(t) = \sum_{m=0}^M \sum_{n=0}^N a_{mn} t^m e^{-\lambda_n t},$$

wobei $a_{mn} \in \mathbb{R}$ und $\lambda_n \in [0, \infty)$ sei. Diese Einschränkung der verwendbaren Schaltzeitverteilungen ist durch die Anwendbarkeit effizienter numerischer Lösungsverfahren begründet [32]. Viele gebräuchliche Verteilungen wie die Dreiecksverteilung, deterministische, exponentielle und andere sind Spezialfälle. Beliebige Verteilungen können approximiert werden, wobei zwischen der Genauigkeit der Anpassung und dem dadurch verursachten Berechnungsaufwand abgewogen werden muß. Um als Modellierer eine bessere Vorstellung von der Form der Schaltzeitverteilung zu haben, wurde für die Spezifikation die Wahrscheinlichkeitsdichtefunktion *pmf* anstatt der *PDF* gewählt. Die Syntax für die Spezifikation der Schaltzeitverteilung enthält Anhang C.

Unter diesen Voraussetzungen ist der dem Modell zugrundeliegende stochastische Prozeß vom Typ **semi-regenerativ** [20]. Es gibt dann sogenannte Regenerationszeitpunkte, an denen der stochastische Prozeß gedächtnislos ist, seine weitere Entwicklung also nur vom augenblicklichen Zustand abhängt. Für diese Zeitpunkte wird eine eingebettete Markow-Kette (*embedded markov chain*, EMC) gebildet. Wenn zwischen den Regenerationszeitpunkten ein Zustandsübergang mit nichtexponentieller Schaltzeit aktiviert ist, muß eine zugeordnete Markow-Kette (*subordinated markov chain*, SMC) zur Berechnung der Übergangswahrscheinlichkeiten und mittleren Aufenthaltsdauern der EMC in diesem Zustand gelöst werden. Nach der Berechnung des eingeschwungenen Zustands der eingebetteten Markow-Kette wird mit Hilfe einer Matrix von Konversionsfaktoren die Lösung des ursprünglichen stochastischen Prozesses bestimmt.

Bestimmung der Zustandswahrscheinlichkeiten

Der im folgenden vorgestellte Algorithmus zur Berechnung der Aufenthaltswahrscheinlichkeiten der andauernden Zustände im eingeschwungenen Zustand (*steady-state*) des Modells entspricht dem in [32] für *extended DSPN* vorgestellten. Der Einfachheit halber wird dabei vom „Schalten eines Übergangs“ in einem Zustand gesprochen, obwohl zur Angabe eines Zustandsübergangs (Schrittes) eigentlich auch die Angabe der Transition notwendig ist, zu der der Übergang gehört.

Zunächst wird der reduzierte Erreichbarkeitsgraph berechnet, wie oben im Abschnitt 4.3.1 beschrieben. Daraus werden folgende Informationen benötigt:

- $Z^{exp} \subseteq Z^{tim}$, die Menge der Zustände, in denen nur Übergänge mit exponentieller Schaltzeit schaltfähig sind (auch als exponentielle Zustände bezeichnet),
- $\forall z_i \in Z^{exp}, z_k \in Z^{tim} : \lambda_{z_i, z_k}$, die Schaltrate des Übergangs mit exponentieller Schaltzeit von Zustand z_i nach Zustand z_k ,
- U^{gen} , die Menge der Übergänge mit nichtexponentiell verteilter Schaltzeit,

- $\forall u \in U^{gen} : Z^u, Q^u, \Delta^u, u^{delay}$. Dabei sind Z^u die Zustände, in denen der Übergang u schaltfähig ist. Mit Q^u wird die Matrix der exponentiellen Schaltraten von Zustandsübergängen bezeichnet, die parallel zum schaltfähigen Übergang u in den Zuständen aus Z^u schaltfähig sind. $\Delta^u = [\delta_{z_i, z_k}]$ ist eine Matrix, die für jeden Zustand $z_i \in Z^u$ die Wahrscheinlichkeit dafür enthält, daß nach dem Schalten des nichtexponentiellen Übergangs in z_i durch einen Pfad zeitloser Übergänge der Zustand z_k erreicht wird. u^{delay} beschreibt die Schaltzeitverteilung des nichtexponentiellen Übergangs.

Aufgrund der Bedingung, daß pro Zustand maximal ein nichtexponentieller Übergang aktiviert sein darf, sind die Mengen Z^{exp} und Z^u für alle $u \in U^{gen}$ paarweise disjunkt, und ihre Vereinigung ergibt wieder Z^{tim} , die Menge der andauernden Zustände.

Im nächsten Schritt müssen folgende Matrix-Integralgleichungen für alle $u \in U^{gen}$ gelöst werden:

$$\begin{aligned}\Omega^u &= \int_0^\infty e^{Q^u t} d u^{delay}(t) \\ \Psi^u &= \int_0^\infty e^{Q^u t} (1 - u^{delay}(t)) dt\end{aligned}$$

Dabei ist Ω^u die Matrix der Zustandsübergangswahrscheinlichkeiten des zugeordneten stochastischen Prozesses (*subordinated process*) am Ende der Aktivierung von u und Ψ^u die Matrix der mittleren Aufenthaltszeiten (*expected sojourn times*) in den Zuständen des zugeordneten Prozesses, während u aktiviert ist. Die Zeilen der Matrizen Ψ^u und Ω^u müssen nur für die initialen Zustände des zugeordneten Prozesses bestimmt werden.

Nun muß die Matrix \mathcal{P} der einzelnen Zustandsübergangswahrscheinlichkeiten (*one-step transition probability matrix*) der eingebetteten Markow-Kette und die Matrix \mathcal{C} der Konversionsfaktoren berechnet werden:

$$\begin{aligned}\forall z_i \in Z^{exp} : \\ \mathcal{P}_{z_i, z_k} &= \begin{cases} 0 & \text{falls } i = k \\ \frac{\lambda_{z_i, z_k}}{\lambda_{z_i}} & \text{sonst} \end{cases} \\ \mathcal{C}_{z_i, z_k} &= \begin{cases} \frac{1}{\lambda_{z_i}} & \text{falls } i = k \\ 0 & \text{sonst} \end{cases} \\ \forall u \in U^{gen}, \forall z_i \in Z^u : \\ \mathcal{P}_{z_i} &= \Omega_{z_i}^u \Delta^u \\ \mathcal{C}_{z_i, z_k} &= \begin{cases} \Psi_{z_i, z_k}^u & \text{falls } z_k \in Z^u \\ 0 & \text{sonst} \end{cases}\end{aligned}$$

wobei λ_{z_i} die Summe aller vom Zustand z_i abgehenden Raten exponentieller Übergänge bezeichnet.

Nun kann der Vektor der mittleren Aufenthaltswahrscheinlichkeiten γ der eingebetteten Markov-Kette im eingeschwungenen Zustand berechnet werden, indem folgendes lineares Gleichungssystem gelöst wird:

$$\gamma(\mathcal{P} - \mathcal{I}) = 0, \quad \sum_i \gamma_i = 1$$

Dabei bezeichnet \mathcal{I} die Einheitsmatrix geeigneter Dimension.

Um den Vektor $\pi \in \mathbf{R}^{|Z^{tim}|}$ der mittleren Aufenthaltswahrscheinlichkeiten der andauernden Zustände des stochastischen Prozesses zu erhalten, muß γ mit der Matrix \mathcal{C} der Konversionsfaktoren multipliziert und normalisiert werden:

$$\gamma' = \gamma \mathcal{C}, \quad \pi = \frac{1}{\sum_i \gamma'_i} \gamma'$$

π_i gibt dann die Wahrscheinlichkeit dafür an, daß sich der aus dem Modell ergebende stochastische Prozeß zu einem beliebigen Zeitpunkt im andauernden Zustand $z_i = (r_i, PV_i)$ befindet.

Berechnung der Leistungsparameter

Nun können die Werte der Leistungsparameter im eingeschwungenen Zustand bestimmt werden. Dabei steht $expr^{\mathbf{B}} : R \rightarrow \mathbf{IB}$ für einen markierungsabhängigen booleschen Ausdruck und dementsprechend $expr^{\mathbf{R}} : R \rightarrow \mathbf{IR}$ für einen markierungsabhängigen reellwertigen Ausdruck. Diese Ausdrücke können wie im Abschnitt 4.1 beschrieben zur Spezifikation von Leistungsparametern verwendet werden.

Die Werte von Leistungsparametern werden dann wie folgt berechnet:

- Wahrscheinlichkeit eines booleschen Ausdrucks $expr^{\mathbf{B}}$:

$$P[expr^{\mathbf{B}}] = \sum_{i: expr^{\mathbf{B}}(r_i)} \pi_i$$

- Wahrscheinlichkeit eines booleschen Ausdrucks $expr_1^{\mathbf{B}}$ unter der Bedingung, daß Ausdruck $expr_2^{\mathbf{B}}$ gilt:

$$P[expr_1^{\mathbf{B}} | expr_2^{\mathbf{B}}] = \frac{\sum_{i: expr_1^{\mathbf{B}}(r_i) \wedge expr_2^{\mathbf{B}}(r_i)} \pi_i}{\sum_{i: expr_2^{\mathbf{B}}(r_i)} \pi_i}$$

- Erwartungswert eines reellwertigen Ausdrucks $expr^{\mathbf{R}}$:

$$E[expr^{\mathbf{R}}] = \sum_{i=1}^{|Z^{tim}|} expr^{\mathbf{R}}(r_i) \pi_i$$

- Erwartungswert eines reellwertigen Ausdrucks $expr^{\mathbf{R}}$, wobei nur Zustände beachtet werden, in denen der boolesche Ausdruck $expr^{\mathbf{B}}$ gilt:

$$E[expr^{\mathbf{R}} | expr^{\mathbf{B}}] = \frac{\sum_{i: expr^{\mathbf{B}}(r_i)} expr^{\mathbf{R}}(r_i) \pi_i}{\sum_{i: expr^{\mathbf{B}}(r_i)} \pi_i}$$

- Der Durchsatz einer zeitbehafteten Transition $t \in T^{tim}$ wird aus der Summe der Durchsätze der einzelnen Übergänge bestimmt:

$$TP[t] = \sum_{u \in U(t)} \sum_{i: (t,u) \in \Theta_{z_i}^*} \overline{u^{delay}}(r_i) \pi_i$$

Dabei bedeutet $\overline{u^{delay}}(r_i)$ die durchschnittliche Schaltrate des Übergangs u in der Markierung r_i . Der so berechnete Durchsatz ist für Transitionen mit nichtexponentiellen Übergängen, die durch das Schalten eines anderen Übergangs deaktiviert werden können, nicht korrekt. Das liegt daran, daß die Schaltzeit einer solchen Transition nicht gedächtnislos ist wie die Exponentialverteilung.

Der größte Berechnungsaufwand ist dabei für die Lösung der Integralgleichungen für Ω^u und Ψ^u notwendig. In [32] werden dafür effiziente Lösungsverfahren vorgeschlagen. Um die Darstellung der Vorgehensweise bei der numerischen Analyse zu vereinfachen, wurde hier auf die Betrachtung markierungsabhängiger Schaltraten verzichtet. Wie aus der Definition der speziellen Petri-Netze ersichtlich, ist deren Spezifikation jedoch ausdrücklich erlaubt; die entsprechenden Analyseverfahren aus [32] werden angewendet. Die Semantik eines zeitbehafteten Übergangs mit markierungsabhängiger Schaltzeit ist dabei wie folgt: während der Übergang schaltfähig ist, kann sich durch das Schalten eines anderen Übergangs die Markierung ändern. Dann verändert sich auch die Schaltzeit des Übergangs, wobei der relative Anteil der bereits abgelaufenen Zeit erhalten bleibt. Die Form der Schaltzeitfunktion ändert sich nicht. Dies wird als *marking dependence through a scaling factor* bezeichnet [3].

4.3.3 Simulation

In manchen Fällen ist die direkte numerische Analyse nicht anwendbar, zum Beispiel wenn der Erreichbarkeitsgraph zu groß wird oder die strukturellen Einschränkungen der verwendeten Netzklasse verletzt werden. Dann können die angegebenen Leistungsparameter simulativ berechnet werden. Dieser Abschnitt erläutert die dabei verwendeten Techniken, die in [42] für farblose Petri-Netze vorgestellt wurden und hier entsprechend angepaßt eingesetzt werden. Unter Simulation des Modells wird die automatische Untersuchung des Verhaltens (das Markenspiel oder *token game*) als diskrete Ereignissimulation verstanden. Ein zufälliger Pfad des stochastischen Prozesses wird dabei durchgespielt und beobachtet, so daß eine mit statistischen Methoden auswertbare Stichprobe der Leistungsparameterwerte ermittelt werden kann.

Bei der numerischen Analyse wird das gesamte Verhalten des Modells berechnet, welches durch den Erreichbarkeitsgraph gegeben ist, und daraus dann die interessierenden Werte abgeleitet. Prinzipiell können also aus einer einmal berechneten Lösung später weitere Aussagen bestimmt werden. Im Unterschied dazu werden bei der Simulation nur die angegebenen Leistungsparameter während des Simulationslaufs beobachtet. Das Ende des Simulationsexperiments ist erreicht, wenn alle Leistungsparameter mit der geforderten Genauigkeit bestimmt wurden, so daß die Laufzeit eines Experiments im Gegensatz zur Analyse stark von der Wahl der Leistungsparameter abhängt.

Während die statistisch abgesicherte Bestimmung der Leistungsparameter aus einem Simulationslauf komplexer als ihre Berechnung aus einer numerisch erhaltenen analytischen Lösung ist, müssen innerhalb des Modells weniger Beschränkungen als bei der Analyse beachtet werden. Da der Erreichbarkeitsgraph und sich daraus ergebende Matrizen nicht aufgestellt werden, sind keine größeren Anforderungen an den verfügbaren Hauptspeicher zu stellen. Insbesondere lassen sich auch manche Modelle mit unendlich großem Zustandsraum untersuchen. Darüber hinaus sind auch Modelle ohne Probleme handhabbar, in denen mehrere Übergänge mit nichtexponentiell verteilter Schaltzeit in einem Zustand aktiviert sind. Die zwei größten Probleme der numerischen Analyse gelten damit für Simulationsexperimente nicht. Dafür ist hier die Laufzeit eines Simulationsexperiments und die gesicherte Bestimmung der Leistungsparameter aus dem Simulationsverlauf problematisch. Beide Techniken sind darum als einander ergänzende Methoden zur Lösung derselben Aufgabe anzusehen, wobei jede ihre spezifischen Vorteile unter bestimmten Voraussetzungen hat. Die numerische Analyse sollte angewendet werden, solange es aufgrund der Einschränkungen möglich ist, da sie gesicherte Ergebnisse oft schneller als die Simulation ermittelt.

Mit den genannten Problemen der Simulation beschäftigt sich [42] für farblose stochastische Petri-Netze der Klasse *extended DSPN*. Die dabei erzielten Ergebnisse lassen sich zum Teil ohne weiteres auf die hier verwendete spezielle farbige Netzklasse übertragen, so daß eine effiziente Simulation mit statistisch abgesicherter Bestimmung der Leistungsparameter möglich ist. Mit der Realisierung eines Simulationswerkzeugs zur Leistungsbewertung der speziellen Petri-Netze beschäftigt sich Abschnitt 7.2.

Bestimmung der Leistungsparameter durch Simulation

Ausgehend vom initialen Zustand z_0 wird während eines Simulationslaufs jeweils die Menge der schaltfähigen Übergänge bzw. Schritte bestimmt und nach den Schaltregeln einer davon ausgewählt, der dann schaltet und zu einem neuen Zustand führt. Für neu schaltfähig gewordene Transitionen muß dabei eine Schaltzeit zufällig, aber entsprechend ihrer Verteilungsfunktion ausgewählt werden. Die Vorgehensweise entspricht dem im Abschnitt 4.3.1 beschriebenen Netzverhalten. Im Gegensatz zur numerischen Analyse muß dabei nur der aktuelle Zustand aufbewahrt werden. Handelt es sich dabei um einen andauernden Zustand, bleiben die aktuellen Werte der Leistungsparameter für die Dauer dieses Zustands stabil. Die Werte der Leistungsparameter und die Zustandsdauer sind Eingangsdaten der Algorithmen zur Schätzung der Leistungsparameter und ihrer bisher erreichten Genauigkeit.

Es werden dieselben Arten der Definition von Leistungsparametern verwendet wie bei der numerischen Analyse (siehe Seite 66). Demnach sind wieder Erwartungswerte reellwertiger Ausdrücke $expr^{\mathbf{R}}$ und Wahrscheinlichkeiten boolescher Ausdrücke $expr^{\mathbf{B}}$ im eingeschwungenen Zustand zu berechnen. Während eines Simulationsexperimentes entsteht so für jeden Leistungsparameter eine Abfolge von Paaren $(v_{i,expr}, t_i)$ für $i = 1 \dots n$, wobei t_i die Dauer des i -ten Zustands $z_i = (r_i, PV_i)$ ist und $v_{i,expr}$ der folgendermaßen definierte Wert des Ausdrucks $expr$ in diesem Zustand:

$$v_{i,expr^{\mathbf{R}}} = expr^{\mathbf{R}}(r_i)$$

$$v_{i,expr^{\mathbf{B}}} = \begin{cases} 1 & \text{falls } expr^{\mathbf{B}}(r_i) = \text{Wahr} \\ 0 & \text{sonst} \end{cases}$$

Außerdem sei t_{end} die gesamte abgelaufene Simulationszeit, d.h. $t_{end} = \sum_{i=1}^n t_i$. Da die Einzelwerte der während der Simulation ermittelten Größen identisch verteilt sind, kann der Erwartungswert der Leistungsparameter durch den arithmetischen Mittelwert geschätzt werden. Demnach ist

- die Wahrscheinlichkeit eines booleschen Ausdrucks $expr^{\mathbf{B}}$:

$$P[expr^{\mathbf{B}}] = \frac{1}{t_{end}} \sum_{i=1}^n t_i v_{i,expr^{\mathbf{B}}}$$

- die Wahrscheinlichkeit eines booleschen Ausdrucks $expr_1^{\mathbf{B}}$ unter der Bedingung, daß Ausdruck $expr_2^{\mathbf{B}}$ gilt:

$$P[expr_1^{\mathbf{B}} | expr_2^{\mathbf{B}}] = \frac{\sum_{i=1}^n t_i v_{i,expr_1^{\mathbf{B}}} v_{i,expr_2^{\mathbf{B}}}}{\sum_{i=1}^n t_i v_{i,expr_2^{\mathbf{B}}}}$$

- der Erwartungswert eines reellwertigen Ausdrucks $expr^{\mathbf{R}}$:

$$E[expr^{\mathbf{R}}] = \frac{1}{t_{end}} \sum_{i=1}^n t_i v_{i,expr^{\mathbf{R}}}$$

- der Erwartungswert eines reellwertigen Ausdrucks $expr^{\mathbf{R}}$, wobei nur Zustände beachtet werden, in denen der boolesche Ausdruck $expr^{\mathbf{B}}$ gilt:

$$E[expr^{\mathbf{R}} | expr^{\mathbf{B}}] = \frac{\sum_{i=1}^n t_i v_{i,expr^{\mathbf{R}}} v_{i,expr^{\mathbf{B}}}}{\sum_{i=1}^n t_i v_{i,expr^{\mathbf{B}}}}$$

- der Durchsatz einer zeitbehafteten Transition $t \in T^{tim}$ wird aus der Summe der Durchsätze der einzelnen Übergänge bestimmt. Dazu muß gezählt werden, wie oft die Übergänge in der abgelaufenen Simulationszeit geschaltet haben.

$$TP[t] = \sum_{u \in U(t)} \frac{1}{t_{end}} \sum_{i=1}^n \begin{cases} 1 & \text{falls } z_i \xrightarrow{(t,u)} z_{i+1} \\ 0 & \text{sonst} \end{cases}$$

Neben dem Mittelwert, der den tatsächlichen Wert des interessierenden Leistungsparameter beschreibt, ist auch die Varianz wichtig. Aus ihr können Aussagen über die Genauigkeit und Verlässlichkeit eines geschätzten Parameters gemacht werden. Insbesondere kann ein um den Mittelwert liegender Bereich (Konfidenzintervall) berechnet werden, innerhalb dessen sich der geschätzte Parameter mit einer gewissen Wahrscheinlichkeit befindet.

Einfache Schätzverfahren für die Varianz eines gesuchten Maßes gehen davon aus, daß die Einzelwerte identisch verteilt und voneinander unabhängig sind. Die zweite Voraussetzung ist bei der Simulation der hier verwendeten Modelle nicht erfüllt, da der aktuelle Zustand des Petri-Netzes zu einer starken Korrelation der aufeinander folgenden Einzelwerte führt. Für die hier verwendeten Modelle kann die in [42] beschriebene Methode der Spektralanalyse der Varianz verwendet werden. Sie setzt keine unabhängigen Einzelwerte voraus, sondern nur einen schwach stationären Prozeß. Die Analyse der Autokovarianzen wird dabei in den

Frequenzbereich verlagert. Eine Erläuterung des statistischen Hintergrundes kann der genannten Arbeit entnommen werden. Da für die Methode lediglich die aus der Simulation ermittelten Einzelwerte, nicht aber die Modellstruktur verwendet wird, kann sie auch für die Simulation der in dieser Arbeit verwendeten Modelle angewandt werden.

Während der Anfangsphase eines Simulationsexperimentes aus dem initialen Zustand heraus muß zunächst eine gewisse Simulationszeit vergehen, bevor die ermittelten Einzelwerte zur Bestimmung der Leistungsparameter für den eingeschwungenen Zustand verwendet werden können. Diese Zeit wird als Einschwing- oder transiente Phase bezeichnet, und ihre Beachtung ist wichtig, da ansonsten möglicherweise das Ergebnis verfälscht wird. Sie kann ermittelt werden, indem ein heuristisches grafisches Verfahren sowie ein anschließender Test auf Stationarität durchgeführt wird [42]. Damit steht ein robustes Verfahren zur Verfügung, das gut innerhalb eines automatisierten Simulationswerkzeugs verwendbar ist.

Neben der Bestimmung von Leistungsparametern im eingeschwungenen Zustand können mit der Simulation auch Aussagen über den Zustand des Modells nach einer bestimmten Simulationszeit gemacht werden. Bei dieser transienten Simulation wird ein Simulationsexperiment (oder mehrere parallel ablaufende) immer wieder beim initialen Zustand begonnen und bis zum spezifizierten Zeitpunkt durchgeführt. Die Untersuchung der Ergebnisse ist statistisch einfacher, da sie nicht nur identisch verteilt, sondern auch voneinander unabhängig sind und demnach mit Standardverfahren behandelt werden können. Bei der transienten Simulation wird natürlich die transiente Phase nicht verworfen, denn es sollen ja gerade Aussagen über das Einschwingverhalten gemacht werden. Sind bis zum Erreichen des transienten Zeitpunktes viele Ereignisse zu simulieren, kann die transiente Simulation aber sehr lange Berechnungszeiten bis zum Erreichen einer angegebenen Genauigkeit benötigen.

Beschleunigung von Simulationsläufen

Neben der statistisch fundierten Bestimmung der Leistungsparameter aus den Einzelwerten ist die lange Dauer eines Simulationslaufs oft störend oder macht die Untersuchung in der zur Verfügung stehenden Zeit undurchführbar. Viele Arbeiten beschäftigen sich darum mit einer Beschleunigung von Simulationsexperimenten. Von den in [42] beschriebenen und im Softwarewerkzeug TimeNET [33] realisierten Methoden ist die parallele Simulation ohne Probleme für die in dieser Arbeit vorkommenden Modelle anwendbar. Dabei wird nicht das Modell selbst aufgeteilt, wie es in vielen anderen Arbeiten vorgeschlagen wird, da dies Probleme bezüglich der Bestimmung sinnvoller Teilmodellgrenzen aufwirft und nur dann sinnvoll ist, wenn der Kommunikationsaufwand zwischen den Teilmodellen gering ist im Vergleich zum Simulationsaufwand der Modelle selbst. Die auftretenden Modelle passen in den Hauptspeicher der verwendeten Arbeitsplatzrechner, so daß eine Aufteilung des Modells nicht zwingend notwendig ist.

Statt dessen werden verschiedene Replikationen eines Simulators, die alle das vollständige Modell simulieren, gestartet. Eine Synchronisation zwischen den Replikationen ist dann nicht notwendig. Durch die Verwendung unterschiedlicher Startwerte für die eingesetzten Zufallszahlengeneratoren sind die aus verschiedenen Replikationen stammenden Einzelwerte unabhängig, was ihre Untersuchung vereinfacht. Mit einem Netz von 15 Arbeitsrechnern kann so eine Beschleunigung um den Faktor drei bis fünf erreicht werden [42].

Kapitel 5

Funktionssymbole als alternatives Beschreibungsmittel

Mit den speziellen Petri-Netzen, die in den vorangegangenen Kapiteln beschrieben wurden, lassen sich Fertigungsprozesse komfortabel modellieren. Viele Anwender von Werkzeugen zur Modellierung von Fertigungssystemen verwenden aber zur Zeit andersartige, weiter verbreitete Beschreibungsmittel als Petri-Netze. In der Produktionstechnik haben sich eigene Symbol- und Modellwelten für dieses Anwendungsgebiet entwickelt, und formalen Ansätzen wie Petri-Netzen wird oftmals mit einer gewissen Skepsis begegnet. Um dieses Problem zu überwinden, wird in diesem Kapitel eine der Produktionstechnik entlehnte Modellierungsart zusätzlich zu Petri-Netzen angeboten. Dabei werden als Beschreibungsmittel die bereits in Kapitel 2.5 angesprochenen Funktionssymbole verwendet. Die entstehenden Modelle können in ein spezielles Petri-Netz umgewandelt und danach mit den vorgestellten Techniken bewertet werden.

Mit der Ausnutzung der in der Modulbibliothek der speziellen Petri-Netze vorhandenen Informationen für die symbolische Modellierung beschäftigt sich der erste Abschnitt des Kapitels. Darauf aufbauend wird gezeigt, wie das fiktive Beispiel aus Abschnitt 2.2 mit Hilfe der Funktionssymbole modelliert werden kann. Die Vorgehensweise bei der automatischen Umwandlung der entstehenden Modelle in ein spezielles Petri-Netz erläutert der letzte Abschnitt.

5.1 Verwendung der Modulbibliothek

Die verwendeten Funktionssymbole orientieren sich an den im Softwarewerkzeug MOSYS [71, 55] eingesetzten Modellierungselementen. Diese umfassen sechs verschiedene Grundelemente mit jeweils eigenem grafischen Symbol für Bearbeiten, Lagern, Transportieren, Montieren, Demontieren sowie Systemein- und Ausgang. Die weiteren Eigenschaften eines Symbols werden in beiden Ansätzen über Parameter angegeben, wobei in der vorliegenden Arbeit noch zwischen Untertypen eines Grundsymbols unterschieden wird.

Bild 5.1 zeigt ihre hier verwendete grafische Darstellung. Das Prinzip der getrennten Modellierung von Struktur und Arbeitsplänen wird im Gegensatz zu MOSYS auch bei symbolischen

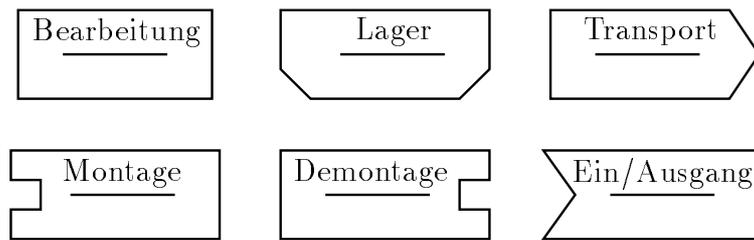


Abbildung 5.1: Verwendete Funktionssymbole

Modellen angewendet. Die Linie in der Mitte jedes Symbols trennt zwischen dem Namen des Betriebsmittels (oben, Struktur) und des Arbeitsgangs (unten, Arbeitsplan). Häufig werden sowohl Montage- als auch Demontagevorgänge durch ein Betriebsmittel (Spannplatz etc.) durchgeführt. Im Strukturmodell werden darum keine Symbole für Montage oder Demontage verwendet, sondern an ihrer Stelle das Symbol Bearbeiten. Im Arbeitsplanmodell werden dann die genaueren Funktionssymbole eingesetzt.

Nach der Erstellung eines symbolischen Modells soll dieses in ein spezielles Petri-Netz umgewandelt werden. Da jedes Funktionssymbol für ein Betriebsmittel bzw. einen Arbeitsgang steht, ist ihre Ersetzung durch Modelle aus der in Abschnitt 3.3 beschriebenen Modulbibliothek naheliegend. Die Übersetzung erfolgt dann durch Ersetzen der Symbole durch parametrisierte Bausteine und wird im Abschnitt 5.3 dargestellt. Jedes Grundsymbol (z.B. Transport) steht für eine Gruppe verschiedener Betriebsmittel dieses Typs, die mit unterschiedlichen Bibliotheksmodellen abgebildet werden (Förderband, fahrerloses Transportsystem u.a.). Die Menge der verfügbaren Unterarten eines Symbols richtet sich nach den entsprechend definierten Teilmodellen und kann erweitert werden. Die Menge der Grundsymbole ist ebenfalls nicht prinzipiell beschränkt.

Die Verwaltung der Bausteinbibliothek erfolgt wie bereits im Abschnitt 3.3 beschrieben grafisch durch die eingesetzte Benutzungsoberfläche. Auf der obersten Seite der Bausteinbibliothek sind die Grundsymbole dargestellt. Jedes dieser Grundsymbole steht wieder für eine Unterseite, die dann Symbole einzelner Bibliotheksmodule enthält. So werden die Module in Gruppen geteilt und den Grundsymbolen zugeordnet. Alle für die symbolische Modellierung verwendbaren Bibliotheksmodule sind mit einem Attribut entsprechend gekennzeichnet.

Ein Gruppensymbol enthält den Namen der Gruppe (z.B. Transport) und ist mit der grafischen Standarddarstellung der Gruppenmitglieder sichtbar. Zwischen den Gruppensymbolen können gerichtete Kanten gezogen werden, die die Verbindungsregeln festlegen. So wird bestimmt, welche Symbole im Modell verbunden werden dürfen; jede Kante steht für eine erlaubte Verbindungsart. Es ist beispielsweise nicht gestattet, zwei Lager direkt miteinander zu verbinden, zumindest ein Transportvorgang muß dazwischen liegen. Bei der Verbindung der eingesetzten Modellteile müssen später Stellen miteinander verschmolzen werden. Dazu muß festgelegt werden, welche Stelle dabei erhalten bleibt und mit ihren Attributen die Eigenschaften der übrigbleibenden Stelle bestimmt. Dies wird mit einem Attribut jeder Verbindungskante zwischen zwei Grundsymbolen angegeben, beispielsweise bleibt die ein Lager modellierende Stelle bei einer Verbindung mit anderen Modellteilen immer erhalten.

Ein für die symbolische Modellierung verwendbares Bibliotheksmodul besitzt als zusätzliche Information eventuell ein von der Standarddarstellung der Gruppe abweichendes Erschei-

oder falsch wird. Die Transition **Montieren** wird nur für den Hauptarbeitsgang benötigt, wogegen die Transition **LadeZusatz** ausschließlich im Nebenarbeitsgang aktiviert wird. Mit den Ein- und Ausgangsbedingungen der Anschlußstellen wird außerdem spezifiziert, unter welchen Bedingungen sie Ein- bzw. Ausgang des Teilmodells sind. Dies geschieht wieder abhängig vom Arbeitsgang. Im Hauptarbeitsgang ist die Stelle **PMontage** sowohl Eingangs- als auch Ausgangsstelle. Im Nebenarbeitsgang ist dagegen die Stelle **EinZusatz** Eingangsstelle, während es keine Ausgangsstelle gibt.

Wenn der Benutzer nun ein symbolisches Modell erzeugen will, wird durch die Benutzungsoberfläche die Bibliothek nach Modellen durchsucht, die als dafür verwendbar gekennzeichnet sind. Die Grundsymbole werden dann in der Menüleiste angezeigt, und es kann jeweils das gewünschte Untersymbol ausgewählt werden. Anhand der Verbindungsregeln in der Bibliothek werden interne Verbindungsregeln der Oberfläche erzeugt, die nur die angegebenen Verbindungen zwischen Funktionssymbolen gestatten. Jedes Einzelsymbol erhält als Attribute alle Parameter des speziellen Petri-Netz-Modells. Dies geschieht sowohl für die symbolischen Struktur- als auch Arbeitsplanmodelle.

5.2 Symbolische Modellierung von Fertigungssystemen

Im Bild 5.3 ist das symbolische Modell der Struktur des fiktiven Fertigungssystems dargestellt, das in Abschnitt 2.2 beschrieben wurde. Durch die Trennung zwischen Struktur und Arbeitsplänen kann auch hier das Modell in etwa entsprechend des Layouts aufgebaut werden.

Die Knoten dieses Modells sind Funktionssymbole, die jeweils für ein Betriebsmittel stehen. Ihre Namen stehen oberhalb der Trennlinie im Symbol, da es sich hier um ein reines Strukturmodell handelt. Beim Modellieren ist bekannt, welche Grundtypen und Unterarten von Betriebsmitteln abgebildet werden sollen. Das entsprechende Symbol kann anhand der grafischen Unterscheidung und der kurzen Beschreibungen der Modelle ausgewählt werden. Die Kanten sind gerichtete Verbindungen, die nach den Verbindungsregeln zulässig sein müssen und jeweils einen möglichen Materialfluß modellieren. Ebenso wie in einem speziellen Petri-Netz der Struktur können zur Vereinfachung von Parameterfestlegungen o.ä. Definitionen verwendet werden, die als Text im Modell erscheinen.

Auf ähnliche Weise werden die Leistungsparameter definiert, deren Werte durch die Untersuchung bestimmt werden sollen. Ein Leistungsparameter besteht wie im Petri-Netz-Modell aus einem Namen und einem Ausdruck. Der Ausdruck kann im symbolischen Modell aber keine von Stellen abhängenden Werte enthalten, da diese nicht bekannt sind. Es dürfen darum nur Definitionen aus den Teilmodellen verwendet werden. Daraus folgt, daß wichtige Kenngrößen eines Teilmodells mit entsprechenden Definitionen im generischen Struktur- oder Arbeitsplanmodell nach außen hin verfügbar gemacht werden müssen. Ein Beispiel dafür ist die Definition des Füllstands des Zwischenpuffers im Teilmodell Montage, für die auf der Strukturseite des Bibliotheksmodells **Fuell = #PZusatz** definiert wird. Der mittlere Füllstand während einer Auswertung kann dann mit der Definition eines Leistungsparameters **Fuellstand = E[Montage\Fuell]** erfolgen. Dabei können zur Berechnung von Wahrscheinlichkeiten und Erwartungswerten von Ausdrücken die Konstrukte verwendet werden, die auch für die Leistungsmaße der speziellen Petri-Netze verfügbar sind (siehe Abschnitt 4.1).

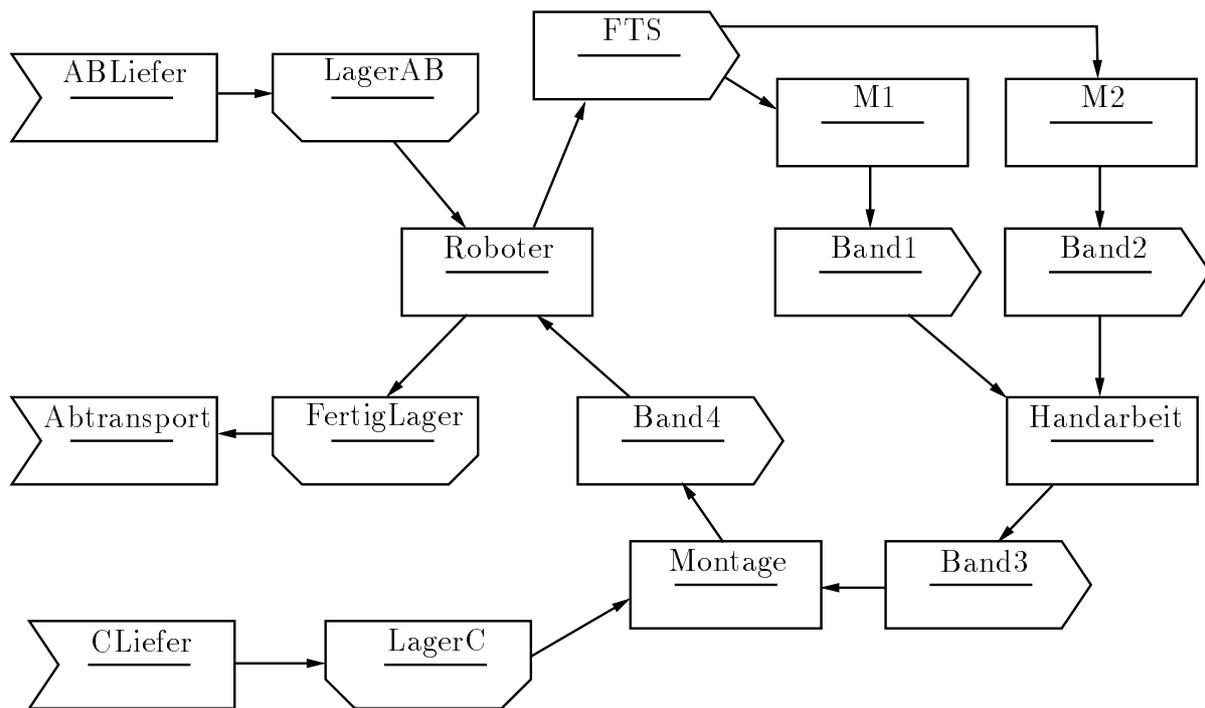


Abbildung 5.3: Symbolisches Strukturmodell

In den Parametern der Symbole werden die strukturellen, d.h. arbeitsplanunabhängigen Eigenschaften wie Ausfallzeit, Kapazität oder normale Bearbeitungsdauer angegeben. Das Bibliotheksmodul **Montage** ist auf Seite 56 dargestellt. Die Parameter **Kap**, **MitAusfall**, **Dauer** und **LadeZeit** können nach dem Anklicken des Funktionssymbols eingestellt werden. Bei der Verwendung in symbolischen Modellen ist der Kontext der Parameter nicht mehr sichtbar, so daß sie von sich aus verständlich sein müssen. Eine Erweiterung der Parameter um einen Kommentar ist darum sinnvoll.

Von den verschiedenen Arbeitsplänen des verwendeten Modells wird hier der Kürze halber nur einer dargestellt. Bild 5.4 zeigt das symbolische Modell des Arbeitsplans A. Jedes Symbol beschreibt einen Arbeitsschritt, dessen Name unter der Trennlinie steht. Im dargestellten Modell ist bereits die Zuordnung von Betriebsmitteln zu Arbeitsschritten erfolgt, die in der Maschinenbelegungsplanung festgelegt wird. Demzufolge ist über der Trennlinie das jeweils verwendete Betriebsmittel eingetragen. Ebenso wie in den assoziierten Petri-Netzen dürfen im Arbeitsplanmodell nur solche Knoten verwendet werden, die Symbolen im Strukturmodell entsprechen. Die Zuordnung erfolgt mit Hilfe der über der Trennlinie stehenden Namen, wobei natürlich der Symboltyp mit der Ausnahme von Montage- und Demontagevorgängen derselbe sein muß.

Die Kanten im Arbeitsplanmodell beschreiben Bewegungen von Werkstücken oder Veränderungen ihres Bearbeitungszustands. An jeder Kante steht der Produktname und der Bearbeitungszustand mit derselben Syntax wie in einem assoziierten Petri-Netz. Es dürfen nur Kanten auftreten, die eine Entsprechung im symbolischen Strukturmodell haben. Davon ausgenommen sind mehrfache Bearbeitungen nacheinander innerhalb derselben Maschine, für die keine Verbindung der Maschine mit sich selbst notwendig ist. Symbolische Arbeitsplanmodelle können ebenfalls aus Name und Ausdruck bestehende Definitionen enthalten.

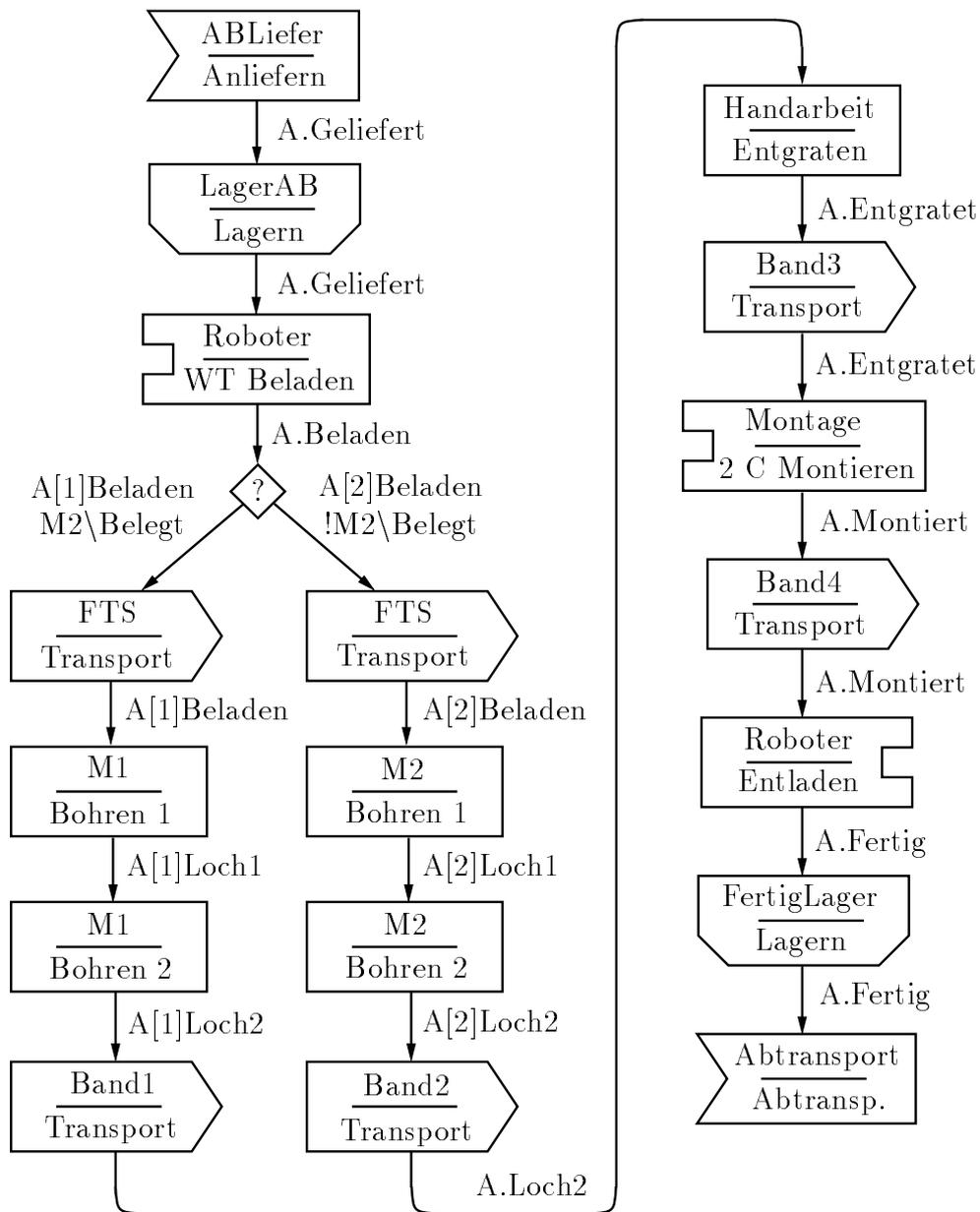


Abbildung 5.4: Symbolisches Arbeitsplanmodell

Normalerweise bestehen symbolische Arbeitsplanmodelle lediglich aus einer linearen Abfolge von Symbolen und Kanten. Die einzige erlaubte Ausnahme stellen Varianten dar, mit denen der Beginn von alternativen Fertigungsvorschriften nachgebildet werden kann. Dazu gibt es den im Bild verwendeten zusätzlichen Verzweigungsknoten. An den Kanten danach stehen die um die Variante in eckigen Klammern erweiterten Produkt- und Zustandsbezeichnungen. Außerdem können diesen Kanten jeweils eine Auswahlwahrscheinlichkeit und -bedingung zugeordnet sein. Die Bedeutung dieser beiden Ausdrücke wurde bereits für die assoziierten Petri-Netze ab Seite 60 beschrieben. Um sinnvolle Ausdrücke bilden zu können, dürfen wie für die Leistungsparameter die in den Bibliotheksmodulen der Symbole auftretenden Definitionen verwendet werden. Im Beispiel hängt die Entscheidung über die zu wählende Variante davon ab, ob die Maschine 2 belegt ist ($[M2\Belegt]$), wobei $Belegt$ eine Definition im Bi-

bibliotheksmodul der Maschine ist. Am Schluß einer Fertigungsvariante enden die Kanten in einem Symbol, und die Variantenbezeichnung wird zurückgesetzt.

Die Parameter der Funktionssymbole im Arbeitsplanmodell beschreiben die Eigenschaften des jeweiligen Arbeitsschritts. In der Montagestation werden hier in einem Arbeitsgang zwei Teile vom Typ C anmontiert. Das Bibliotheksmodell des Arbeitsplans für die Montagestation wurde auf Seite 96 gezeigt. Der Parameter N muß den Wert Zwei erhalten, denn er gibt die Anzahl montierter Teile an. Außerdem muß hier über den Parameter **zusatz** das anzumontierende Werkstück genannt werden. Falls sich die Bearbeitungsdauer vom Standardwert unterscheidet, kann sie dem Parameter **Dauer** zugewiesen werden. Der Nebenarbeitsgang der Montagestation, der die Anlieferung von C-Teilen beschreibt, ist im Arbeitsplan des Werkstücks C enthalten.

Für alle im dargestellten symbolischen Modell verwendeten Funktionssymbole müssen Bibliotheksmodule existieren. Einige davon wurden bereits im Abschnitt 3.3 dargestellt. Die fehlenden Module können aus den speziellen Petri-Netz-Modellen abgeleitet werden, die in der Beschreibung nicht als Bibliotheksmodul realisiert worden waren. In beiden Fällen müssen die notwendigen Anschlußbedingungen und Parameter Eingangs- bzw. Ausgangsmarke zusätzlich spezifiziert werden. Für einfache Modelle mit nur einem Ein- und Ausgang ist die Angabe von Ein- und Ausgangsbedingungen nicht notwendig, da die Zuordnung eindeutig ist. Außerdem müssen die die bearbeiteten Werkstücktypen modellierenden Variablen an den Kanten der Arbeitspläne im Bibliotheksmodell als Parameter vom Typ Eingangs- oder Ausgangsmarke deklariert werden. Diese wurden in den im Abschnitt 3.3 dargestellten Beispielen *vorher* und *nachher* genannt.

5.3 Umwandlung in ein spezielles Petri-Netz

Die im letzten Abschnitt beschriebenen Modelle müssen nun automatisch in ein spezielles Petri-Netz nach Kapitel 3.1.1 umgewandelt werden, um die entwickelten Untersuchungsmethoden anwenden zu können. Das geschieht hauptsächlich, indem für jedes Symbol des Modells ein Teilmodell aus der in Kapitel 3.3 beschriebenen Bibliothek eingesetzt wird. Außerdem müssen noch die nötigen Verbindungen zwischen diesen Teilmodellen erzeugt werden. Dies wird sowohl für das Strukturmodell als auch für die Modelle der Arbeitspläne einzeln durchgeführt. Die Auswahl des richtigen Bibliotheksmodells sowie die Ersetzung der Parameter durch Werte geschieht mit den im symbolischen Modell angegebenen Parametern.

Bevor ein symbolisches Modell in ein spezielles Petri-Netz übersetzt werden kann, ist ein Konsistenztest notwendig. Damit können fehlerhaft spezifizierte Modelle mit einer entsprechenden Fehlermeldung an den Modellierer zurückgewiesen werden. Überprüft wird unter anderem, ob

- es für jedes Symbol und jede Kante im Arbeitsplanmodell eine Entsprechung im Strukturmodell gibt,
- die verwendeten Symbole gültigen Bibliotheksmodulen zugeordnet sind,
- alle im Modul angegebenen Parameter belegt sind,

- die Kanten im Strukturmodell den Verbindungsregeln entsprechen,
- die in Ausdrücken verwendeten Abfragen von Werten der Teilmodelle in diesen definiert sind,
- die Arbeitsplanmodelle bis auf Varianten eindeutig sind,
- und ob an allen Kanten im Arbeitsplanmodell sinnvolle Produktbezeichnungen stehen.

Anschließend kann das symbolische Strukturmodell aus Bild 5.3 in ein spezielles Petri-Netz der Struktur umgewandelt werden. Bei entsprechender Zuordnung von Teilmodellen zu Symbolen wird dieses dem in Bild 3.2 dargestellten Modell gleichen. Im ersten Schritt der Übersetzung wird zunächst eine neue, leere Primärseite für das Strukturmodell erzeugt. Für jedes im symbolischen Modell auftretende Funktionssymbol wird nun das zugeordnete Bibliotheksmodul instanziiert, nachdem die verwendeten Parameter eingesetzt wurden. Die Instanziierung eines Bibliotheksmoduls wird entsprechend der im Abschnitt 3.3 erklärten Methode durchgeführt. Abhängig von den eingegebenen Parametern können demnach nicht nur Zahlenwerte wie Bearbeitungszeiten o.ä. beeinflußt werden, sondern durch die mit Transitionen verknüpften Existenzbedingungen ist auch die Struktur des Modells veränderbar. Im Ergebnis dieses Schrittes existiert auf der Primärseite des neuen Modells für jedes Funktionssymbol eine Vertretertransition mit ihren Sockelstellen. Diese werden zunächst nicht untereinander verbunden, sondern erst später bei der Übersetzung der Arbeitspläne. Das liegt daran, daß erst in den Arbeitsplänen mit den Anschlußbedingungen eindeutige Zuordnungen zwischen den zusammengehörigen Sockelstellen der Vertretertransitionen bekannt sind. Die Bedeutung der Kanten im symbolischen Modell liegt daher lediglich in der Beschränkung der im symbolischen Arbeitsplanmodell verwendbaren Verbindungen. Anschließend werden die Definitionen und Leistungsparameter ohne Veränderung in das Modell übernommen.

Für das Beispiel der Montagestation wird demnach das in Bild 3.11 auf Seite 55 gezeigte Strukturmodell instanziiert. Die Parameter **Kap**, **MitAusfall**, **Dauer** und **LadeZeit** des Modells werden entsprechend den im symbolischen Modell angegebenen Werten belegt. Für das Strukturmodell sind keine zusätzlichen Informationen zum normalen Bibliotheksmodul anzugeben.

Nun wird für jedes symbolische Arbeitsplanmodell ein assoziiertes Petri-Netz-Modell erzeugt. In den dafür angelegten Primärseiten wird wiederum für jedes Funktionssymbol ein Teilmodell aus der Bibliothek eingesetzt, wobei hier natürlich das Arbeitsplanmodell verwendet wird. Bei der Instanziierung werden zunächst die vom Modellierer angegebenen Parameter belegt. Der Wert der Parameter vom Typ Eingangs- und Ausgangsmarke wird automatisch von den mit dem Symbol verbundenen Kanten abgelesen. Wegen der Linearität der Arbeitsplanmodelle ist diese Zuordnung eindeutig. Dabei werden lediglich solche Parameter eingesetzt, die an mit Anschlußknoten verbundenen Kanten im Bibliotheksmodell auftreten. Dies geschieht nach der Auswertung der Existenzbedingungen der Transitionen, so daß es auch für Modelle mit mehreren Ein- oder Ausgängen immer nur höchstens einen gültigen Parameter vom Typ Ein- bzw. Ausgangsmarke gibt. Mit dieser Besonderheit wird dann das Bibliotheksmodul instanziiert, und es existiert auf der Primärseite für jedes Funktionssymbol eine Vertretertransition mit den verbundenen Anschlußstellen. Diese werden nur für Sockelstellen der Unterseite erzeugt, deren Anschlußbedingung wahr ist.

Für Sockelstellen mit erfüllter Eingangsbedingung wird eine Kante von der Anschlußstelle zur Vertretertransition gezogen, für diejenigen mit erfüllter Ausgangsbedingung eine Kante in umgekehrter Richtung. Durch die Linearität der Arbeitsplanmodelle muß dabei immer genau eine Eingangs- und eine Ausgangsstelle auf der Primärseite erzeugt werden, eine Ausnahme bilden lediglich der Beginn und das Ende eines Arbeitsplans. Anhand der Reihenfolge der Arbeitsschritte ist nun eindeutig festgelegt, welche Stellen auf der Primärseite des Arbeitsplanmodells miteinander verschmolzen werden müssen. Durch die Verbindung der Ausgangsstelle einer Vertretertransition mit der Eingangsstelle der nachfolgenden Vertretertransition wird ein durchgehender Arbeitsplan erzeugt. Bei der Verschmelzung zweier Stellen im Arbeitsplan werden die dazugehörigen Stellen im Strukturmodell ebenfalls miteinander verschmolzen, wenn dies nicht bereits geschehen ist. Dabei werden die bereits genannten Verbindungsregeln eingesetzt. Sie legen fest, welche der beiden verschmolzenen Stellen erhalten bleibt und deren Attribute (Name, Kapazität, initiale Markierung) somit übernommen werden.

Mehrere Arbeitsschritte sollten nacheinander innerhalb derselben Maschine nur dann durchgeführt werden, wenn die Werkstücke auf derselben Bearbeitungsstelle liegenbleiben und keine Vorpuffer o.ä. passieren müssen. Ist dies nicht der Fall, kann entweder der Vorpuffer als eigenes Funktionssymbol modelliert werden, oder die zusätzlichen Vorgänge wie Transport in den Vorpuffer mit Existenzbedingungen belegt werden. Diese werden falsch und löschen damit die entsprechenden Transitionen, wenn ein zusätzlicher Parameter „ohne Vorpufferung“ o.ä. vom Modellierer auf Wahr gesetzt wird. Da bereits anhand der Modellstruktur erkennbar ist, ob mehrere Arbeitsgänge in derselben Maschine nacheinander ausgeführt werden, könnten auch Standardparameter dafür eingeführt und diese dann automatisch während der Übersetzung mit Werten belegt werden.

Wenn ein Werkstück innerhalb seines Arbeitsplans eine Bearbeitungsstation ohne Zustandsänderung lediglich passiert, muß dies im generischen Arbeitsplan der Station vorgesehen sein. Dazu kann wieder vom Modellierer die Art des Arbeitsgangs als Parameter verwendet werden, der hier beispielsweise den Wert „keine Bearbeitung“ hätte. Abhängig davon werden durch Existenzbedingungen die nicht benötigten Transitionen bei der Übersetzung gelöscht bzw. nicht in das fertige Modell übernommen.

Die Übersetzung der Verzweigungsknoten in das spezielle Petri-Netz ist einfach möglich, da dafür nur eine Verzweigung von Kanten mit der spezifizierten Wahrscheinlichkeit und Bedingung erzeugt werden muß (siehe Abschnitt 3.4). Zum Schluß sind noch die im symbolischen Modell angegebenen Definitionen in das spezielle Petri-Netz zu übernehmen.

Im Beispiel der Montagestation (generischer Arbeitsplan siehe Bild 5.2) muß für den dargestellten Arbeitsgang „2 C Montieren“ der Parameter **Arbeitsgang** auf den Wert „Haupt“ eingestellt werden. Darüber hinaus erhalten die Parameter folgende Werte: **N** wird auf Zwei und **zusatz** auf C gesetzt, da dies Anzahl und Art der montierten Teile sind. Durch die Auswertung der Existenzbedingungen werden die Transition **LadeZusatz** und die verbundenen Kanten nicht übernommen. Der Parameter **zusatz** vom Typ Eingangsmarke wird dadurch nicht mehr bei der automatischen Belegung mit Produktbezeichnungen berücksichtigt. Die beiden übrigbleibenden Parameter des Typs Eingangs- bzw. Ausgangsmarke **vorher** und **nachher** werden aufgrund der Kantenbezeichnungen im symbolischen Modell automatisch mit den Werten **A.Entgratet** und **A.Montiert** belegt. Nach der Instanziierung des Bibliotheksmodells werden auf der Primärseite des Arbeitsplanmodells neben der Vertretertransi-

tion zwei Sockelstellen erzeugt. Diese sind beide mit der Ein- und Ausgangsstelle **PMontage** der Unterseite verbunden und werden im weiteren Verlauf der Übersetzung mit der vorangehenden bzw. nachfolgenden Stelle auf der Primärseite verschmolzen.

Das Ergebnis der Übersetzung sind spezielle Petri-Netze der Struktur und der Arbeitspläne. Diese ähneln für das Beispiel zwar stark den durch die direkte Modellierung als Petri-Netze entstandenen Modellen aus Kapitel 3, sind aber zumindest in der grafischen Darstellung nicht identisch. Entscheidend ist aber letztendlich, daß das modellierte Verhalten dasselbe ist. Bei der Platzierung der Elemente des Petri-Netzes können die Positionen der ersetzten Symbole verwendet werden. Damit kann der Modellierer das Ergebnis der Übersetzung ansehen, obwohl das nicht notwendig ist. Die weitere Untersuchung des Modells kann so erfolgen, daß der Benutzer nicht mit dem erzeugten Petri-Netz in Berührung kommt. Die fertig übersetzten Modelle werden dann mit den in Kapitel 3 und 4 beschriebenen Methoden zu einem Gesamtmodell zusammengefaßt und danach untersucht. Nach der quantitativen Bewertung sind die Ergebnisse als Werte der definierten Leistungsparameter ablesbar. Schwieriger ist eine Untersuchung und Darstellung der qualitativen Ergebnisse, da diese für das symbolische Modell kaum interpretierbar sind. Aus der Lebendigkeit und Beschränktheit des erzeugten Petri-Netz-Modells kann aber auch auf entsprechende Eigenschaften des symbolischen Modells geschlossen werden.

Zusammenfassend läßt sich einschätzen, daß die Modellierung mit Funktionssymbolen lediglich dann sinnvoll ist, wenn der Modellierer bereits mit diesem Beschreibungsmittel gearbeitet hat und sich nicht mit Petri-Netzen beschäftigen will. Allerdings ist auch dann eine Einarbeitung notwendig, da die vorgeschlagene Methode einige für die Umsetzung in Petri-Netze notwendige Eigenheiten aufweist. Trotz der prinzipiellen Verwendbarkeit der Bibliotheksmodule für die Umwandlung eines symbolischen Modells ist zusätzlicher Aufwand zu ihrer entsprechenden Vorbereitung notwendig. Bei der Modellierung mit Funktionssymbolen ist die Ausdrucksfähigkeit und die Untersuchungsmethodik eingeschränkt.

Die bei der Übersetzung eines symbolischen Modells in ein spezielles Petri-Netz auftretenden Probleme weisen auf Defizite in der Eindeutigkeit eines Modells mit Funktionssymbolen hin. Insbesondere ist die Unterscheidung von aktiven und passiven Ressourcen eines Fertigungssystems wie Maschinen und Lagern nicht vollständig möglich, da beispielsweise auch Maschinen oft Pufferplätze enthalten. Darüber hinaus sind die Verbindungen zwischen Symbolen nicht eindeutig, wenn diese mehrere Ein- oder Ausgänge besitzen. Die Abbildung dieser Details ist in einem Petri-Netz direkt möglich.

Kapitel 6

Ein Anwendungsbeispiel

Dieses Kapitel beschreibt die Modellierung und anschließende Untersuchung eines realen Fertigungssystems mit der in dieser Arbeit entwickelten Modellierungs- und Analysemethode. Anhand dessen wird ihre praktische Anwendbarkeit bewertet. Vorarbeiten dazu sind in [87, 24] enthalten. Die darin entwickelten Modelle wurden für die folgende Darstellung weitgehend überarbeitet. Auf eine Modellierung mit Hilfe der in Kapitel 5 dargestellten Funktionssymbole wird verzichtet.

6.1 Beschreibung des Fertigungssystems

Das als Anwendungsbeispiel gewählte Fertigungssystem wird durch einen Hersteller von Produkten der Fahrzeuginsassensicherheit betrieben. Es handelt sich um eine Montagelinie für pyrotechnische Gurtstrammer. In entsprechend ausgerüsteten Fahrzeugen ist das Schloß des Sicherheitsgurtes an einem solchen Gurtstrammer befestigt, so daß im Falle einer Kollision der Gurt um einige Zentimeter gespannt werden kann. Dadurch wird der Fahrzeuginsasse beim Aufprall weniger weit nach vorn geschleudert. Zur Herstellung eines Gurtstrammers müssen 20 verschiedene Teile montiert werden, wobei einige Montagevorgänge recht komplex sind. Aufbau und Funktionsweise eines Gurtstrammers sind in [24] näher erläutert.

Die Modellierung und Bewertung erfolgte in Zusammenarbeit mit dem Hersteller der Montagelinie, einer auf die Entwicklung und den Aufbau hochautomatisierter Produktionsanlagen spezialisierten mittelständischen Anlagenbaufirma. Zum Zeitpunkt der Untersuchung der Montagelinie war diese bereits seit kurzer Zeit in Betrieb. Dies hatte für die Anwendung der Modellierungstechnik den Vorteil, daß die Eigenschaften der einzelnen Betriebsmittel gemessen bzw. aus vorhandenen Mitschreibungen gewonnen werden konnten. Außerdem wurden die Ergebnisse der Leistungsbewertung des modellierten Systems mit den realen Werten verglichen, um das Modell und die Untersuchungsmethoden zu überprüfen.

Die Motivation für die Untersuchung ergab sich daraus, daß die Montagelinie nicht die ursprünglich geplante Leistungsfähigkeit erreichte. Als Grund dafür wurde vermutet, daß Ausfälle und Störungen der einzelnen Stationen und ihre Auswirkungen auf den Durchsatz während des Entwurfs nicht ausreichend beachtet worden waren. Dies sollte anhand eines

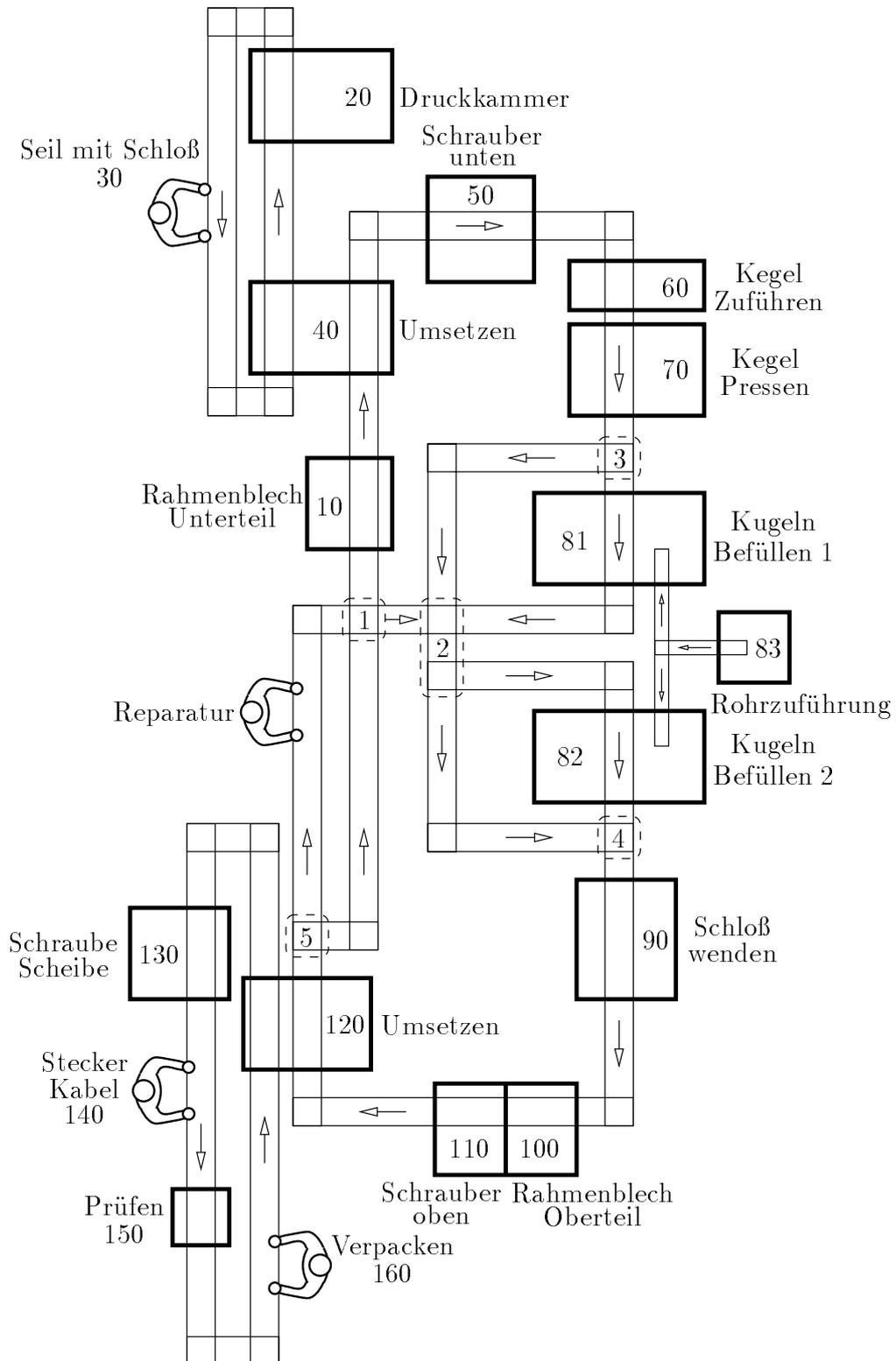


Abbildung 6.1: Aufbau der Montagelinie

Modells überprüft werden. Außerdem wurde eine Untersuchung verschiedener Einflußfaktoren auf die Produktionsleistung angestrebt. Darüber hinaus waren Vorschläge zu entwickeln, wie die Leistungsfähigkeit der Montagelinie ohne größere Investitionen erhöht werden kann.

Aufgrund des Kostendrucks und der Absatzerwartungen ist die Montagelinie weitgehend automatisiert. Lediglich vier der insgesamt achtzehn Stationen sind als Handarbeitsplatz ausgelegt. Bild 6.1 skizziert den Aufbau des Fertigungssystems sowie die Lage der einzelnen Betriebsmittel. Fett gezeichnete Rechtecke stehen für automatische Montagestationen, an denen die jeweilige Nummer und eine Kurzbeschreibung der Funktion eingezeichnet ist. Die vier Handarbeitsplätze werden ebenfalls als Fertigungsstationen aufgefaßt, die bis auf den Reparaturplatz auch durch Nummern gekennzeichnet sind. Der Transport der Werkstücke findet ausschließlich auf Werkstückträgern statt, die über Transportbänder fortbewegt werden. An den Transportbändern ist jeweils die Bewegungsrichtung markiert. Sie transportieren die auf ihnen liegenden Werkstückträger kontinuierlich vorwärts, bis diese durch einen vor ihnen liegenden Werkstückträger blockiert werden oder in der Zielstation angekommen sind. Die Transportbänder haben außerdem die Funktion von Puffern zwischen den Montagestationen. Die Pufferkapazität ist durch die Länge des Bandes und der eines Werkstückträgers bestimmt.

Zur Verringerung des Einflusses der erfahrungsgemäß größeren Schwankungen in der Bearbeitungszeit an Handarbeitsplätzen wurde der gesamte Montageablauf auf drei Montagekarrees aufgeteilt. Das Hauptmontagekarree enthält bis auf den Reparaturplatz nur automatisierte Stationen, während die manuellen Tätigkeiten am Vor- und Endmontagekarree durchgeführt werden. In jedem Montagekarree wird ein anderer Typ Werkstückträger eingesetzt. Diese unterscheiden sich hauptsächlich in der Anzahl der Nester, in denen sich die Gurtstrammer befinden. Die Werkstückträger des Vormontagekarrees besitzen acht, die des Hauptmontagekarrees zwei und die des Endmontagekarrees vier Nester.

Um die Werkstückträger im Hauptmontagekarree zur jeweiligen Folgestation transportieren zu können und verschiedene Teilströme wieder zusammenzufassen, wurden Weichen in die Bandstrecke eingefügt. Diese sind im Bild 6.1 gestrichelt hervorgehoben und numeriert. Die Weichen 2, 3 und 4 sind für die alternative Bearbeitung der Bauteile an den Stationen 81 und 82 notwendig. Die Weiche 2 faßt vereinfachend zwei reale Weichen zusammen, bei denen nur an der unteren eine Entscheidung zwischen verschiedenen Zielen möglich ist. An der Weiche 5 werden Werkstückträger mit fehlerhaften Bauteilen zum Reparaturplatz, leere dagegen direkt zur Station 10 gesendet. Weiche 1 wird für nur die Einschleusung reparierter Teile benötigt.

Vormontagekarree

Tabelle 6.1 nennt die im Vormontagekarree auftretenden Bearbeitungszustände der Gurtstrammer auf dem hier verwendeten Werkstückträger. Ein Werkstückträger wird durch seinen Namen (WTV) sowie einer durch einen Punkt getrennten Kurzbezeichnung des Zustands beschrieben.

In Station 20 des Vormontagekarrees werden Gasgenerator und Dichtring in die Druckkammer eingefügt und diese dann auf einen leeren Werkstückträger gelegt. Wenn dies für alle

Bezeichnung	Bearbeitungszustand bzw. Inhalt des Werkstückträgers
WTV.leer	leerer Werkstückträger (8 Nester)
WTV.Druck	8 Druckkammern
WTV.Schl8	8 Schloßpeitschen und Druckkammern
WTV.Schl6	6 Schloßpeitschen und Druckkammern
WTV.Schl4	4 Schloßpeitschen und Druckkammern
WTV.Schl2	2 Schloßpeitschen und Druckkammern

Tabelle 6.1: Bearbeitungszustände im Vormontagekarree

Nester geschehen ist, wird der Werkstückträger zur Folgestation 30 transportiert. Dort werden die Schloßpeitschen manuell in die Druckkammern eingefädelt. Die vormontierten Teile werden durch die Station 40 auf Werkstückträger des Hauptmontagekarrees umgesetzt. Danach fahren die leeren Werkstückträger wieder zur Station 20. Tabelle 6.2 enthält den entsprechenden Arbeitsplan.

Nr	Funktion	WTV.? vor/nach	Betriebsmittel	
1	Transport zu Station 20	leer	leer	Band vor 20
2	WT mit Druckkammern füllen	leer	Druck	Station 20
3	Transport zu Station 30	Druck	Druck	Band vor 30
4	Schloßpeitschen einlegen	Druck	Schl8	Station 30
5	Transport zu Station 40	Schl8	Schl8	Band vor 40
6	2 Teile entnehmen und umsetzen	Schl8	Schl6	Station 40
7	2 Teile entnehmen und umsetzen	Schl6	Schl4	Station 40
8	2 Teile entnehmen und umsetzen	Schl4	Schl2	Station 40
9	2 Teile entnehmen und umsetzen	Schl2	leer	Station 40

Tabelle 6.2: Arbeitsplan für Werkstückträger im Vormontagekarree

Hauptmontagekarree

Tabelle 6.3 enthält eine Liste der im Hauptmontagekarree auftretenden Bearbeitungszustände der Werkstücke auf dem hier verwendeten Werkstückträger (WTH).

Bezeichnung	Bearbeitungszustand bzw. Inhalt des Werkstückträgers
WTH.leer	leerer Werkstückträger (2 Nester)
WTH.Rahmen	untere Rahmenbleche aufgelegt
WTH.Schl	Schloßpeitsche mit Druckkammer auf Rahmenblech
WTH.SchrU	unteres Rahmenblech verschraubt
WTH.Kegel	Verriegelungskegel aufgefädelt
WTH.Press	Verriegelungskegel verpreßt
WTH.Rohr	Kugeln eingefüllt und Druckrohr montiert
WTH.Rolle	Umlenkrolle eingelegt, Werkstück gewendet
WTH.SchrO	oberes Rahmenblech montiert und verschraubt

Tabelle 6.3: Bearbeitungszustände im Hauptmontagekarree

An Station 10 des Hauptmontagekarrees werden Rahmenbleche auf leere Werkstückträger gesetzt, auf die in Station 40 jeweils eine der vormontierten Schloßpeitschen mit Druckkammer gelegt werden. Das untere Rahmenblech wird in Station 50 mit der Druckkammer verschraubt. In den Stationen 60 und 70 wird ein Verriegelungskegel auf die Schloßpeitsche aufgefädelt und auf ihr durch Pressen befestigt. An den Stationen 81 und 82 werden Kugeln um den Kegel gelegt und in das Druckrohr geschoben, welches danach mit der Druckkammer verschraubt wird. Da dieser Vorgang sehr zeitaufwendig ist, werden zwei identische Stationen für diese Aufgabe eingesetzt. Werkstückträger werden von Station 70 zu einer der beiden Stationen 81 oder 82 geschickt, so daß jede nur den halben Durchsatz des restlichen Systems erreichen muß. Dieses Verhalten wird im tabellarischen Arbeitsplan mit der Angabe eines Fertigungszweigs beschrieben, der als Zahl die Nummer des Arbeitsgangs ergänzt. Es wird jeweils nur eine Alternative für ein Werkstück gewählt. Sie endet, wenn kein Fertigungszweig mehr angegeben ist. Geschachtelte Varianten sind auf diese Weise nur schwer beschreibbar.

Nr/Zw	Funktion	WTH.? vor/nach		Betriebsmittel
1	Transport zu Station 10	leer	leer	Band W1-10
2	untere Rahmenbleche auflegen	leer	Rahmen	Station 10
3	Transport zu Station 40	Rahmen	Rahmen	Band vor 40
4	Schloßpeitschen vom WTV auflegen	Rahmen	Schl	Station 40
5	Transport zu Station 50	Schl	Schl	Band vor 50
6	unteres Rahmenblech anschrauben	Schl	SchrU	Station 50
7	Transport zu Station 60	SchrU	SchrU	Band vor 60
8	Verriegelungskegel auffädeln	SchrU	Kegel	Station 60
9	Transport zu Station 70	Kegel	Kegel	Band vor 70
10	Verriegelungskegel verpressen	Kegel	Press	Station 70
11	Transport zu Weiche 3	Press	Press	Band nach 70
12	1 Transport zu Station 81	Press	Press	Band vor 81
13	1 Druckrohr und Kugeln montieren	Press	Rohr	Station 81
14	1 Transport zu Weiche 2	Rohr	Rohr	Band 81-W2
15	1 Transport zu Weiche 4	Rohr	Rohr	Band W2-W4
16	2 Transport zu Weiche 2	Press	Press	Band W3-W2
17	2 Transport zu Station 82	Press	Press	Band vor 82
18	2 Druckrohr und Kugeln montieren	Press	Rohr	Station 82
19	2 Transport zu Weiche 4	Rohr	Rohr	Band 82-W4
20	Transport zu Station 90	Rohr	Rohr	Band vor 90
21	Umlenkrolle einlegen, Teil wenden	Rohr	Rolle	Station 90
22	Transport zu Station 100/110	Rolle	Rolle	Band vor 100
23	oberes Rahmenblech anschrauben	Rolle	SchrO	Station 100/110
24	Transport zu Station 120	SchrO	SchrO	Band vor 120
25	umsetzen der Werkstücke auf WTE	SchrO	leer	Station 120
26	Transport zu Weiche 5	leer	leer	Band 120-W5
27	Transport zu Weiche 1	leer	leer	Band W5-W1

Tabelle 6.4: Arbeitsplan für Werkstückträger im Hauptmontagekarree

In Station 83 werden die zu montierenden Druckrohre mit Etiketten versehen und dann den Stationen 81 und 82 zugeführt. Station 90 legt eine Umlenkrolle ein und wendet das Werkstück. Durch die Stationen 100 und 110 werden die oberen Rahmenbleche aufgelegt und anschließend mit der Druckkammer verschraubt. In Station 120 werden die fehlerlosen Bauteile auf Werkstückträger des Endmontagekarrees umgesetzt. Leere Werkstückträger fahren dann wieder zur Station 10.

Tabelle 6.4 zeigt den Arbeitsplan des Hauptmontagekarrees für Werkstücke ohne Fehler. Die vollständigen Arbeitspläne sind in [24] enthalten. Bedingt durch ihren zusammenhängenden Aufbau werden die Stationen 100 und 110 nicht getrennt betrachtet. Durch verschiedene automatisierte Prüfungen, die innerhalb der Montagestationen stattfinden, werden Produktionsfehler innerhalb des Hauptmontagekarrees erkannt. Die fehlerhaften Teile werden nicht zum Endmontagekarree umgesetzt, sondern nach der Station 120 zum Reparaturplatz geleitet. Der dort beschäftigte Arbeiter sondert Ausschuß aus und bessert reparaturfähige Teile nach. Abhängig vom Bearbeitungszustand nach der Reparatur werden die Werkstücke dann zu einer der Stationen 10, 82 oder 90 geschickt, so daß möglichst wenige Stationen ohne Bearbeitung durchfahren werden müssen. Zu diesem Zweck wurde die Weiche 1 und ihre Verbindung zur Weiche 2 eingebaut. Diese Vorgänge sind nicht im tabellarischen Arbeitsplan enthalten. Im tabellarischen Arbeitsplan sind Weichen mit dem Buchstaben W und ihrer Nummer abgekürzt.

Endmontagekarree

Die Tabelle 6.5 enthält die im Endmontagekarree auftretenden Bearbeitungszustände der Gurtstrammer auf dem hier eingesetzten Werkstückträger (WTE).

Im Endmontagekarree werden durch Station 120 vorgefertigte Gurtstrammer vom Hauptmontagekarree auf leere Werkstückträger gelegt. In Station 130 werden eine Schraube, eine Sicherungsscheibe und ein Etikett angebracht. Am Handarbeitsplatz 140 wird das zur Auslösung des pyrotechnischen Gasgenerators benötigte Zündkabel angesteckt und mit einem Halteclip befestigt. Die korrekte Ausführung dieses Arbeitsgangs wird durch Station 150 überprüft. An der Station 160 werden die fertigen Gurtstrammer von Hand entnommen und verpackt. Leere Werkstückträger werden wieder zur Station 120 transportiert. In Tabelle 6.6 ist der Arbeitsplan des Endmontagekarrees dargestellt.

Bezeichnung	Bearbeitungszustand bzw. Inhalt des Werkstückträgers
WTE.leer	leerer Werkstückträger (4 Nester)
WTE.Teil2	2 vorgefertigte Gurtstrammer
WTE.Teil4	4 vorgefertigte Gurtstrammer
WTE.Etikett	Gurtstrammer mit Schraube, Scheibe und Etikett
WTE.Kabel	Gurtstrammer komplett mit Kabel
WTE.Fertig	geprüfter Gurtstrammer

Tabelle 6.5: Bearbeitungszustände im Endmontagekarree

Nr	Funktion	WTE.? vor/nach		Betriebsmittel
1	Transport zu Station 120	leer	leer	Band vor 120
2	2 Teile auf WTE legen (vom WTH)	leer	Teil2	Station 120
3	2 Teile auf WTE legen (vom WTH)	Teil2	Teil4	Station 120
4	Transport zu Station 130	Teil4	Teil4	Band vor 130
5	Schraube, Scheibe, Etikett montieren	Teil4	Etikett	Station 130
6	Transport zu Station 140	Etikett	Etikett	Band vor 140
7	Zündkabel und Halteclips montieren	Etikett	Kabel	Station 140
8	Transport zu Station 150	Kabel	Kabel	Band vor 150
9	Kabel und Halteclip prüfen	Kabel	Fertig	Station 150
10	Transport zu Station 160	Fertig	Fertig	Band vor 160
11	Gurtstrammer verpacken	Fertig	leer	Station 160

Tabelle 6.6: Arbeitsplan für Werkstückträger im Endmontagekarree

Bestimmung der relevanten Daten

In den vorherigen Abschnitten wurden der strukturelle Aufbau und die Arbeitspläne der Montagelinie beschrieben. Daneben sind Aussagen über die Bearbeitungs- und Transportzeiten, Puffergrößen sowie das Ausfallverhalten für die Modellerstellung notwendig. Die genauen Daten können jeweils [24] entnommen werden.

Für die Bearbeitungszeiten der Stationen wurden während des laufenden Betriebes Meßwerte aufgenommen, deren Mittelwert und Varianz bestimmt wurde. Erwartungsgemäß variiert die Bearbeitungszeit einer automatischen Station kaum, während die Varianz der Handarbeitsplätze im Bereich der Bearbeitungszeit liegt.

Weiterhin mußten die Transportzeiten für einen Werkstückträger auf den verschiedenen Transportbandabschnitten gemessen werden. Dies geschah unter der Voraussetzung, daß die Teilstrecke leer war. Die Transportzeit hängt dann ausschließlich von der konstanten Geschwindigkeit der Transportbänder ab, so daß die Meßwerte keine nennenswerte Varianz aufwiesen. Daneben wurde aus der Länge der Bandabschnitte jeweils die maximale Aufnahmefähigkeit für Werkstückträger bestimmt. Ein weiterer wichtiger Einflußfaktor ist die Menge der eingesetzten Werkstückträger pro Montagekarree, von denen sich insgesamt 82 im Umlauf befinden.

Es wurde untersucht, an welchen Stationen und wie oft Werkstücke fehlerhaft bearbeitet werden. Der Anteil der Ausschussteile daran wurde ebenfalls bestimmt. Insgesamt sind zwischen drei und vier Prozent der gesamten Produktionsleistung Ausschuß. Der Einfluß der Produktionsfehler auf die Gesamtleistung ist demnach bekannt und nicht entscheidend. Um den Ausstoß an Gurtstrammern durch eine Verringerung der Produktionsfehler zu erhöhen, wären technologische Veränderungen an den Stationen notwendig. Die gemessene Verteilung der fehlerhaften Bearbeitungsschritte durch die Stationen könnte Ausgangspunkt für eine entsprechende Fehleranalyse sein. Dies war jedoch nicht Aufgabe der durchgeführten Untersuchung. Fehlerhafte Teile werden darum in den Arbeitsplänen und dem Modell nicht beachtet. Die in [24] für fehlerhafte Teile aufgestellten Arbeitspläne zeigen, daß der erhöhte Modellierungsaufwand nicht durch die geringe Verbesserung der Genauigkeit des Modells gerechtfertigt ist.

Der Aufgabenstellung folgend wurde dagegen dem Ausfallverhalten der automatischen Stationen große Aufmerksamkeit geschenkt. Eine Schätzung der Ausfallzeiten aus Messungen mit einer gewissen statistischen Sicherheit wäre durch den dazu erforderlichen Zeitraum kaum manuell durchführbar gewesen. Durch das Steuersystem der Anlage wird jedoch ein Störungsprotokoll geführt, dessen Datenmaterial ausgewertet wurde. Das typische Verhalten einer automatischen Station ist durch zwei Störungsarten gekennzeichnet. Relativ häufig treten kurze Verklemmungen auf, die zu einer Verzögerung der Bearbeitung führen. Zu ihrer Behebung ist oft kein menschlicher Eingriff nötig, und sie werden im folgenden als Unterbrechung bezeichnet. Seltener sind längere Ausfälle eines Automaten, die z.B. durch den Austausch eines defekten Verschleißteiles ausgelöst werden können. Diese werden als Reparatur bezeichnet und erfordern einen größeren Eingriff. Zur Behebung auftretender Störungen sind Mechaniker verantwortlich, die zusätzlich zu den an den Handarbeitsplätzen tätigen Arbeitern eingesetzt werden.

Aufgrund dieser Vorüberlegungen und der Form der aufgenommenen Kurven wurde vermutet, daß die Verteilungsfunktion der Störungsdauer durch eine gewichtete Summe zweier Exponentialfunktionen nachgebildet werden kann. Die bestimmenden Faktoren dieser Verteilungsfunktionen werden in [24] aus den vorhandenen Werten geschätzt. Ein Anpassungstest zeigt, daß damit auf einfache Weise eine ausreichend genaue Annäherung der realen Störungsdauern erreicht wird. Die Zeitabstände zwischen zwei Störungen werden als exponentialverteilt angenommen.

6.2 Modellierung mit speziellen Petri-Netzen

In diesem Abschnitt wird die vorher beschriebene Montagelinie mit einem speziellen Petri-Netz modelliert. Dies geschieht ohne den Umweg über die alternativ verwendbaren Modelle aus Funktionssymbolen.

Um den Rahmen dieser Arbeit nicht zu sprengen, werden nicht alle Modellteile dargestellt und erläutert. Insgesamt besteht das Modell aus ungefähr 90 Seiten für Arbeitspläne und Struktur, deren größter Teil aber aus Bibliotheksmodulen ableitbar ist. Außerdem wird nicht auf Details der Modellierungstechnik eingegangen, die bereits im Kapitel 3 erklärt wurde. Zur Vereinfachung des Verständnisses der einzelnen Modelle wurden folgende Konventionen eingeführt:

- automatische Stationen und Handarbeitsplätze werden durch eine Transition modelliert, die mit einem **S** und der Stationsnummer benannt ist.
- Die Orte auf dem Transportband, an denen Werkstücke bearbeitet werden können, gehören jeweils zu einer Station. Jede dieser Stellen wird mit einem **P** und der Nummer der zugehörigen Station benannt und hat die Kapazität Eins. Falls es mehrere solcher Stellen gibt, wird zusätzlich ein Buchstabe angehängt.
- Transportbänder werden durch Vertretertransitionen abgebildet. Ihr Name setzt sich aus einem **B** und einer Zahl zusammen, die im Normalfall die Nummer der Zielstation des Bandes angibt.

- An Weichen im Transportsystem wird entschieden, welchen Weg ein Werkstückträger nimmt. Auf ihnen kann sich maximal ein Werkstückträger befinden. Sie werden demnach als Stelle mit der Kapazität Eins modelliert, und ihr Name setzt sich aus einem **W** und einer Zahl zusammen.

Strukturmodell

Bild 6.2 zeigt die Primärseite des Strukturmodells für das Anwendungsbeispiel. Es wurde unter Verwendung der genannten Konventionen aus der Skizze des Aufbaus (siehe Seite 105) abgeleitet. Auf dieser Beschreibungsebene wurden nur Kanten und Stellen benötigt, die Werkstücke bewegen bzw. lagern, und die demnach dick gezeichnet sind. Um die Verständlichkeit des Modells zu erhöhen, wurde es möglichst entsprechend dem Layout gestaltet.

Da bei der Bearbeitung in einer Station der Zustand, nicht aber der Ort eines Werkstücks verändert wird, sind die für die Stationen stehenden Vertretertransitionen durch einen Doppelpfeil mit ihrem jeweiligen Bearbeitungsplatz verbunden. Diese sind keine zusätzliche Art von Kanten, sondern entstehen durch die Darstellung von Kanten in beiden Richtungen. Da das Verhalten der automatischen Stationen und Transportbänder genauer als nur durch eine Transition beschrieben werden soll, wurden jeweils Vertretertransitionen verwendet. Im Gegensatz dazu gibt es bei Handarbeitsplätzen keine Ausfälle und Reparaturen. Die Stationen 30, 140 und 160 werden darum lediglich mit einer zeitbehafteten Transition modelliert. Die Schaltzeit dieser Transition wird exponentialverteilt gewählt, da die Bearbeitungszeiten eine entsprechend hohe Varianz aufweisen.

Die Weiche 2 (Stelle **W2**) wurde vereinfachend als eine Stelle mit Kapazität Zwei modelliert. Außerdem wurde die Weiche 5 weggelassen, deren Funktion durch die Bearbeitungsstelle der Station 120 (Stelle **P120b**) mit übernommen wird. Dies hat keinen entscheidenden Einfluß auf das Verhalten, da durch den geringen Abstand zwischen Station 120 und der Weiche keine Pufferkapazität vernachlässigt wird. Die Stationen 100 und 110 wurden mit einer gemeinsamen Vertretertransition modelliert, da sie konstruktiv eng verbunden sind und nur eine gemeinsame Bestimmung der Ausfall- und Reparaturzeiten möglich war. Auf der Unterseite der Transition werden aber beide Bearbeitungszeiten beachtet. Weiter oben wurde bereits erklärt, warum fehlerhafte Werkstücke bei der Modellierung nicht beachtet werden. Der Reparaturplatz **SRep** und die umliegenden Transportbänder könnten demnach im Modell weggelassen werden. Es stellt aber kein Problem dar, diese Modellteile zur Erinnerung an die reale Struktur zu behalten. Sie werden bei der Erzeugung des Gesamtmodells nicht benötigt und haben keinen Einfluß auf das Verhalten.

Modelle der Arbeitspläne

Nach der Modellierung der Struktur werden die Arbeitspläne in Modelle umgesetzt. Dieser Abschnitt beschreibt die Primärseiten der Arbeitspläne. Auf die Unterseiten wird im folgenden Abschnitt bei der Verfeinerung der Vertretertransitionen zusammen mit deren Struktur eingegangen. Entsprechend der Definition der speziellen Petri-Netze müssen im Arbeitsplanmodell die Produktmarken und ihre Bearbeitungszustände benannt werden. Dazu werden die Bezeichnungen verwendet, die bereits bei der Beschreibung der Montagelinie eingeführt wurden.

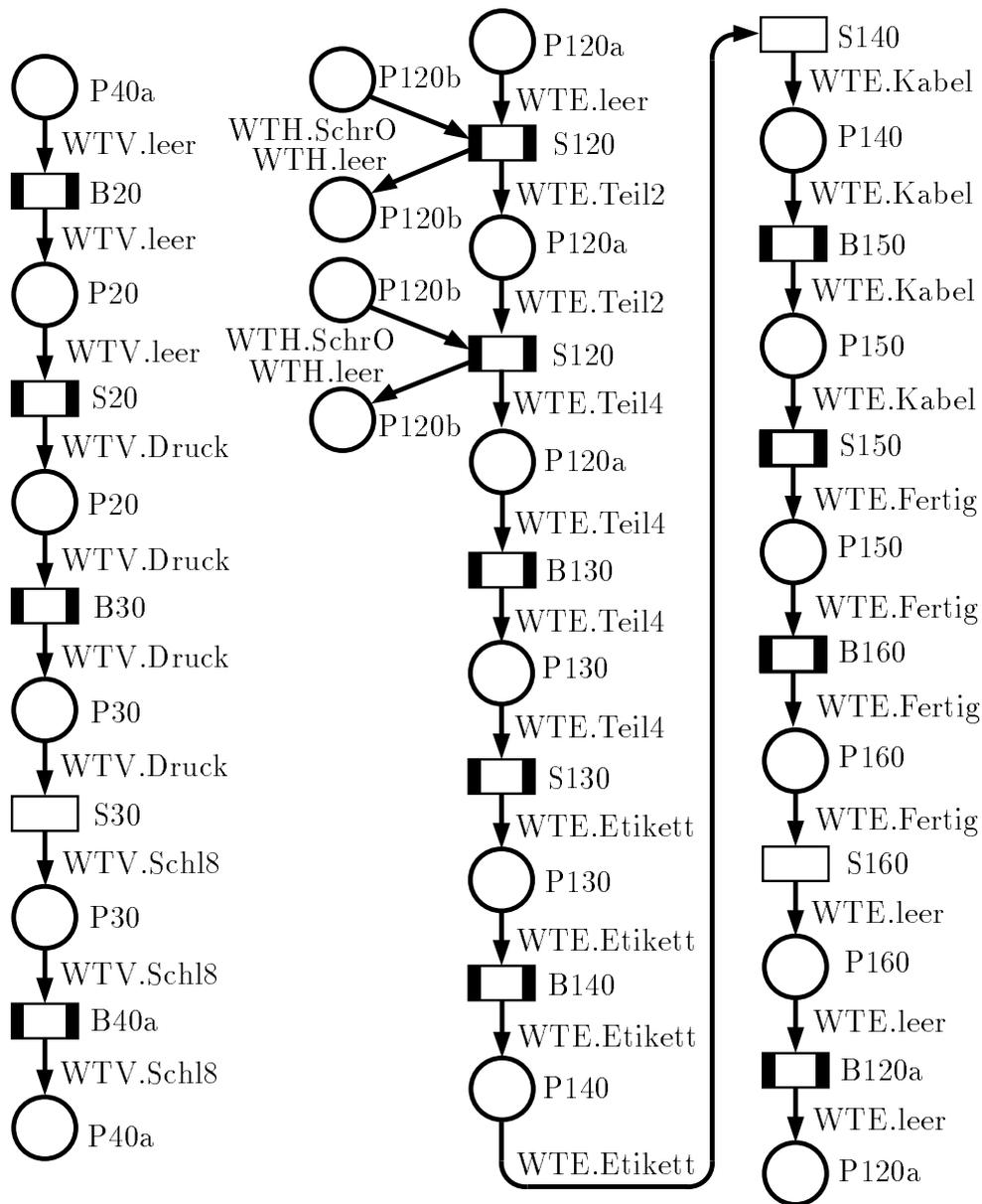


Abbildung 6.3: Arbeitsplanmodelle des Vor- und Endmontagekarrees (Primärseiten)

Die Arbeitspläne beschreiben die in den drei Montagekarrees vorzunehmenden Arbeitsgänge anhand der Werkstückträger und der auf ihnen liegenden Gurtstrammer. Die Anzahl der Werkstückträger ändert sich dabei nicht, und sie unterliegen einem zyklischen Arbeitsplan. Zur besseren Darstellung wurden die Arbeitsplanmodelle an der Stelle aufgetrennt, an der der Werkstückträger leer ist. Sie orientieren sich dadurch an den in Tabellenform angegebenen Arbeitsplänen.

Das Arbeitsplanmodell des Vormontagekarrees ist links im Bild 6.3 dargestellt. Sein einfacher Aufbau entspricht dem linearen Arbeitsplan in Tabelle 6.2. Das Vormontagekarree stellt Werkstückträger mit acht Schloßpeitschen (WTV.Schl8) zur Verfügung, die dann im Hauptmontagekarree weiterverwendet werden. Das Entladen der Werkstückträger wird nicht hier beschrieben, sondern im Arbeitsplan des Hauptmontagekarrees. An den Stationen 20 (und

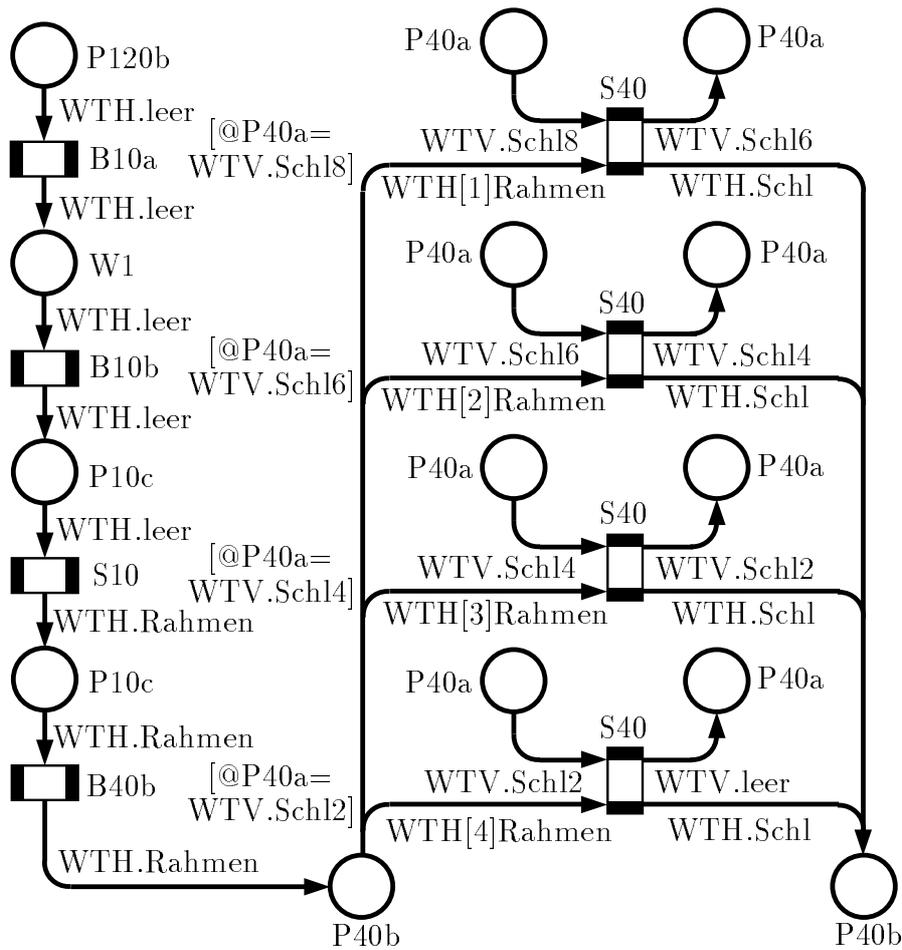


Abbildung 6.4: Arbeitsplanmodell des Hauptmontagekarrees (Primärseite, Teil 1)

10 im Hauptmontagekarree) werden Teile auf leere Werkstückträger gelegt. An diesen Stellen findet eine Einlastung neuer Teile statt. Hier und an allen anderen Montagestationen ist aber stets eine ausreichende Menge zu verarbeitender Teile vorhanden, so daß die Anlieferung keinen Engpaß darstellt. Jeder in einer Station ankommende Werkstückträger wird sofort bearbeitet. Eine Festlegung und Modellierung von Fertigungsaufträgen oder Anlieferung von Teilen ist demnach unnötig.

Der rechte Teil von Bild 6.3 zeigt das Arbeitsplanmodell des Endmontagekarrees. Auch sein Aufbau ist recht einfach, und es konnte mit Hilfe von Tabelle 6.6 erzeugt werden. Zu Beginn des Arbeitsplans werden die vorgefertigten Gurtstrammer von einem Werkstückträger des Hauptmontagekarrees genommen und auf den Werkstückträger des Endmontagekarrees gelegt. Dies wird durch die zusätzlichen Stellen und Kanten an den Transitionen S120 oben im Bild modelliert. Da im Hauptmontagekarree nur zwei, im Endmontagekarree aber vier Gurtstrammer auf einem Werkstückträger liegen, müssen zwei Werkstückträger entladen werden, um einen neuen zu füllen. An Station 160 werden die fertigen Gurtstrammer entnommen und verpackt. Es kann davon ausgegangen werden, daß immer ausreichend Verpackungsmaterial vorhanden ist. Es müssen demnach keine Auslastungsstrategien o.ä. beachtet werden.

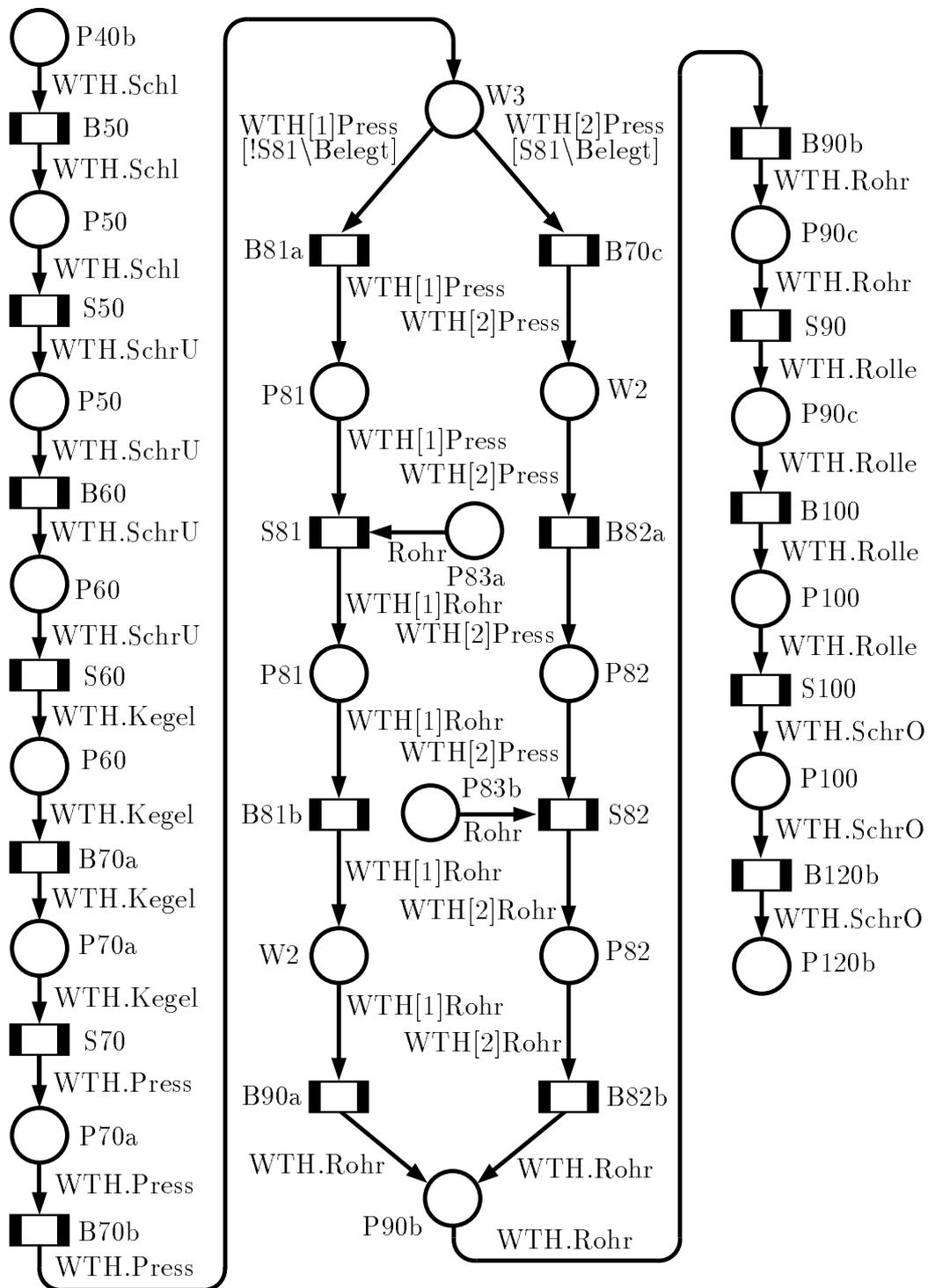


Abbildung 6.5: Arbeitsplanmodell des Hauptmontagekarrees (Primärseite, Teil 2)

Die Primärseite des Arbeitsplanmodells für das Hauptmontagekarree ist recht groß. Sie wurde darum auf die zwei Bilder 6.4 und 6.5 verteilt. Im ersten Teil werden die Arbeitsschritte vom leeren Werkstückträger nach der Entladung durch Station 120 bis zum Auflegen von Schloßpeitschen durch Station 40 beschrieben. Aufgrund der unterschiedlichen Nesteranzahl auf den Werkstückträgern des Vor- und Hauptmontagekarrees müssen in Station 40 vier Fälle unterschieden werden. Je nachdem, ob der Werkstückträger im Vormontagekarree noch acht, sechs, vier oder zwei Schloßpeitschen enthält, wird eine andere Alternative gewählt.

Der interessante Teil des restlichen Modells (Bild 6.5) bildet die Fertigungsvariante an den Stationen 81 und 82 nach. Wenn ein Werkstückträger in Weiche 3 (**W3**) ankommt, wird eine der Alternativen ausgewählt und das Werkstück mit der Nummer der Alternative gekennzeichnet. Im Beispiel ist das für Teile in Weiche 2 (**W2**) wichtig. Ohne eine Kennzeichnung der Alternative wäre hier nicht mehr unterscheidbar, in welche Richtung der Werkstückträger transportiert werden muß. Normalerweise soll Station 81 verwendet werden. Ist diese belegt, wird der Werkstückträger zur Station 82 geschickt. Dies läßt sich mit der Bedingung [**S81**\Belegt] erreichen. Die Transitionen **S81** und **S82** verbrauchen dann bei der Montage die zugelieferten Rohre.

Für leere Werkstückträger ist in diesem Anwendungsbeispiel kein Arbeitsplan notwendig, da sie stets erst gefüllt bzw. bearbeitet werden, bevor sie eine Station verlassen. Die in den Stationen montierten Teile werden fertig angeliefert. Für sie werden demnach auch keine Arbeitspläne benötigt. Lediglich für die Etikettierung der Druckrohre und ihr Transport zu den Stellen **P83a** bzw. **P83a** gibt es einen rudimentären Arbeitsplan, der aber hier nicht dargestellt wird. Die Verwendung der Rohre wird im zweiten Teil des Hauptmontagekarree-Arbeitsplans modelliert.

Verfeinerung der Vertretertransitionen

Viele der dargestellten Vertretertransitionen stehen für ähnliche oder sogar gleiche Strukturen. Für solche immer wieder auftretenden Modellteile wurden Bibliotheksmodule (siehe Abschnitt 3.3) spezifiziert, um die Modellerstellung zu erleichtern.

In Bild 6.6 ist das Bibliotheksmodell der Struktur einer normalen automatisierten Station dargestellt. Dabei wurden zur einfacheren Erläuterung die Schaltzeiten und Schaltgewichte der Transitionen kursiv mit eingezeichnet, die sonst nicht sichtbar sind. Die eigentliche Bearbeitung wird durch die Transition **Maschine** modelliert, die eine deterministische Schaltzeit hat. Nur sie und die mit ihr verbundene Anschlußstelle **Bearb** sind darum dick gezeichnet. Die Stelle **Bearb** ist die Verbindung zum Bearbeitungsplatz auf der Oberseite.

Das im Abschnitt 6.1 beschriebene Ausfall- und Reparaturverhalten wird mit dem oberen elementaren Modellteil nachgebildet. Anfangs ist die Maschine betriebsbereit (eine Marke ist in Stelle **P0K**). Die Zeit bis zu einem Ausfall war als exponentialverteilt angenommen und wird dementsprechend mit der Transition **Ausfall** modelliert, die eine exponentielle Schaltzeit mit der mittleren Schaltzeit *Intakt* hat. Für die Zeit bis zur Behebung des Fehlers wird eine gewichtete Summe aus zwei Exponentialverteilungen verwendet. Diese läßt sich wie im Bild zu sehen durch eine Kombination aus zeitlosen und zeitbehafteten Transitionen modellieren. Die Werte *a* und *b* entsprechen den Gewichten der beiden Exponentialverteilungen, deren Parameter *tau1* und *tau2* als mittlere Schaltzeit der exponentiellen Transitionen

Parameter a Parameter tau1 Parameter DauerBearb
 Parameter b Parameter tau2 Parameter Intakt
 Belegt = (#Bearb > 0)

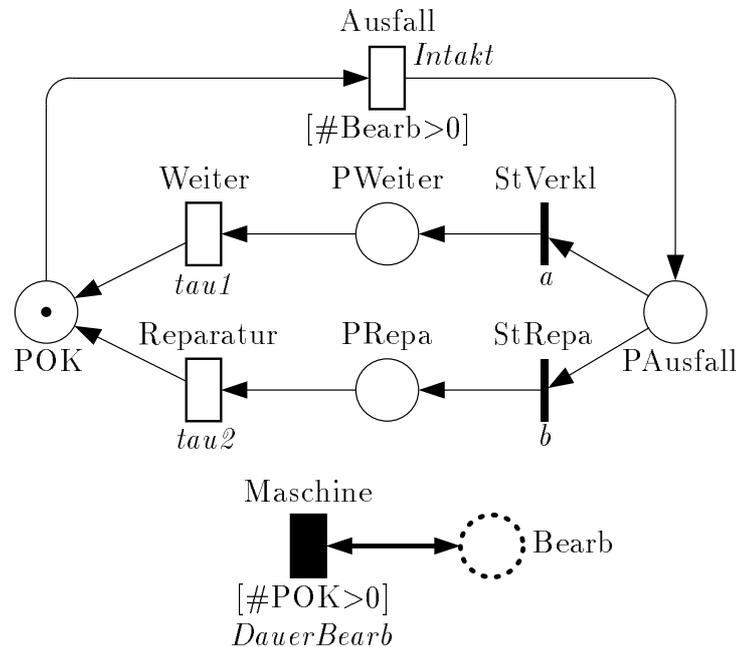


Abbildung 6.6: Strukturmodell einer automatischen Station

Weiter und **Reparatur** eingesetzt werden. Die Maschine kann nur dann arbeiten, wenn sie funktionstüchtig ist. Dies wird mit der Schaltbedingung $[\#POK > 0]$ erreicht.

Über dem grafischen Teil des Modells sind die Parameter des Modells angegeben, die bei der Instanziierung mit Werten belegt werden müssen. Mit ihnen können die noch unbekannt Zahlenwerte der verschiedenen genannten Zeiten und Wahrscheinlichkeiten angegeben werden. Das beschriebene Modell kann für alle automatischen Stationen bis auf 40, 81, 82, 83 und 120 verwendet werden. Außerdem muß von außen auf den Zustand der Station zugegriffen werden können. Dafür wird der Ausdruck **Belegt** definiert.

Der generische Arbeitsplan des Modells einer automatisierten Station beschreibt die Zustandsänderung eines Werkstücks durch die Bearbeitung. Es ähnelt dem auf Seite 55 gezeigten Arbeitsplanmodell einer einfachen Maschine. Für das hier betrachtete Anwendungsbeispiel wird im Unterschied dazu aber keine Schaltbedingung an der Transition **Ausfall** angegeben, da die Ausfallzeiten nicht abhängig von der Belegung der Stationen bestimmt wurden. Die verschiedenen Stationen müssen neben der eigentlichen Montageaufgabe die zu montierenden Teile vereinzeln und vorbereiten. Auch während dieser Zeit kann ein Ausfall stattfinden, selbst wenn sich kein Werkstückträger im Bearbeitungsplatz befindet.

Die Modelle der Stationen 40 und 120, die Werkstücke zwischen den Montagekarrees umsetzen, können aus dem beschriebenen Modell einer automatisierten Station abgeleitet werden. Diese Stationen sind mit jeweils zwei Sockelstellen verbunden, die Bearbeitungsstellen der beiden Montagekarrees entsprechen. Auf der Unterseite des Modells befindet sich eine Transition **Umsetzen**, die mit den beiden dazugehörigen Anschlußstellen verbunden ist. In der

Unterseite des assoziierten Arbeitsplanmodells wird der Zustand der für Werkstückträger stehenden Marken in beiden Stellen verändert. Dies modelliert das Entnehmen bzw. Auflegen von Werkstücken. Das Ausfall- und Reparaturverhalten entspricht dem der normalen Stationen und wird genauso modelliert.

Die Station 83 etikettiert die Druckrohre und führt sie den Stationen 81 und 82 zu. Pro Arbeitsgang werden zwei Druckrohre bearbeitet, die dann auf die Stellen **P83a** und **P83b** verteilt werden. Diese Stellen haben jeweils eine Aufnahmefähigkeit von 12 Rohren. Das Ausfall- und Reparaturverhalten wird wieder wie für die anderen Stationen modelliert. Außerdem enthält die Unterseite der Vertretertransition **S83** eine Transition, die mit den Anschlußstellen der Stellen **P83a** und **P83b** verbunden ist. Der entsprechende Arbeitsplan legt fest, daß Marken vom Typ **Rohr.fertig** in den genannten Stellen erzeugt werden.

Die Stationen 81 und 82 sind identisch aufgebaut. Sie montieren die von Station 83 gelieferten Druckrohre und befüllen diese mit Kugeln. Ihr Modell ähnelt wiederum dem einer automatischen Station. Auf der Unterseite der Transitionen **S81** und **S82** sind außerdem Sockelstellen zur Verbindung mit der Rohrzulieferung (Stellen **P83a** und **P83b**) vorhanden. Diese sind mit der Transition **Maschine** verbunden. Auf der Unterseite eines dazu assoziierten Arbeitsplanmodells werden bei einem Schalten der Transition **Maschine** zwei Marken vom Typ **Rohr.fertig** entnommen und der Zustand der Marke im Bearbeitungsplatz verändert.

Das Verhalten der hier eingesetzten Transportbänder entspricht dem in Abschnitt 3.3 beschriebenen Förderband. Dementsprechend wird das auf Seite 53 gezeigte Bibliotheksmodell mit dem entsprechenden generischen Arbeitsplan verwendet und hier nicht noch einmal erklärt. Allerdings wird die Schaltzeit der Transition **Transport** als deterministisch festgelegt, da die Transportzeiten nicht variieren. Durch die Verwendung dieses Modells wird von der FIFO-Eigenschaft der Puffer abstrahiert. Im betrachteten Modell werden keine Produktionsfehler beachtet, demnach sind alle auf einem Band befindlichen Werkstücke identisch und eine Unterscheidung ist unnötig. Bei der Parametrisierung der Puffer müssen ihre Kapazität und die Transportzeit für ein Teil angegeben werden.

Auf die Umsetzung der Fragestellungen in Leistungsmaße wird im folgenden Abschnitt 6.3 eingegangen. Nach der Modellierung von Struktur und Arbeitsplänen der Montagelinie wird nun ein Gesamtmodell erzeugt. Im Ergebnis wird das Strukturmodell mit Übergangstabellen angereichert, in denen für jede Transition die möglichen Schaltvorgänge angegeben sind. Wie das geschieht, wurde bereits im Abschnitt 3.4 beschrieben.

6.3 Untersuchung und Bewertung

Je größer das Modell eines Fertigungssystems ist, desto schwieriger ist eine Überprüfung seiner gewünschten Funktionsweise. Eine erste Möglichkeit stellt die interaktive Simulation dar, bei der dem Modellierer der Zustand des Modells angezeigt wird und er dann entscheidet, welche der schaltfähigen Transitionen bzw. Übergänge schalten sollen. Dann wird der erreichte Folgezustand dargestellt. Diese Vorgehensweise läßt sich mit der Fehlersuche in einem Programm mit Hilfe von schrittweiser Ausführung vergleichen. Für den Nachweis bestimmter Eigenschaften eines größeren Modells sind jedoch Methoden und Werkzeuge notwendig, die ohne Eingriffe des Modellierers arbeiten. Kapitel 4.2 beschreibt einige davon für die speziellen Petri-Netze.

Qualitative Eigenschaften

Der Nachweis qualitativer Eigenschaften des Modells der Montagelinie wurde durchgeführt, um das erstellte Modell zu überprüfen. Da das modellierte System bereits einige Zeit arbeitet, sind Rückschlüsse auf die Funktionstüchtigkeit der Montagelinie unnötig. Es geht vielmehr um eine Korrektur von Fehlern, die bei der Modellierung eines Fertigungssystems dieser Größenordnung leicht unterlaufen können.

Die Anzahl der Werkstückträger in den drei Montagekarrees ist während der Fertigung konstant. Da jeder Werkstückträger durch eine Marke modelliert wird, muß sich dieser Sachverhalt in entsprechenden Stellen-Invarianten wiederfinden lassen. Darum wurden mit Hilfe der im Abschnitt 4.2 beschriebenen Methode die minimalen Stellen-Invarianten des Modells der Montagelinie bestimmt. Diese werden mit den zu erwartenden Invarianten verglichen, wobei Unterschiede Fehler im Modell erkennen ließen. Neben den drei Invarianten, die die konstante Anzahl von Werkstückträgern in den Montagekarrees beschreiben, wurden folgende Invarianten gefunden: Für das Ausfall- und Reparaturverhalten aller automatisierten Stationen gilt auf der jeweiligen Unterseite $\#POK(e) + \#PAusfall(e) + \#PWeiter(e) + \#PREpa(e) = 1$. Jede dieser Stationen ist also zu einem beliebigen Zeitpunkt in genau einem der möglichen Zustände.

Demnach ist für alle Stellen im Modell die Anzahl der in ihnen befindlichen Marken durch Invarianten beschränkt, außer den Stellen **P83a** und **P83b**. Diese besitzen jedoch eine Kapazität, so daß das gesamte Modell beschränkt ist und einen endlichen Zustandsraum haben muß. Die Lebendigkeit des Modells wird nachgewiesen, indem der Erreichbarkeitsgraph erzeugt wird. Da er keine absorbierenden Zustände enthält, ist das Modell zumindest in dem Sinn lebendig, daß in jedem Zustand mindestens ein Übergang schalten kann.

Konflikte zwischen Transitionen bzw. ihren Übergängen weisen auf Entscheidungen hin, die in bestimmten Zuständen zwischen mehreren möglichen Aktivitäten getroffen werden müssen. Dies ist z.B. für Werkstückträger der Fall, die sich in der Weiche 3 befinden. In dem entsprechenden Arbeitsplanmodell wird aber festgelegt, unter welchen Bedingungen einer der möglichen Wege ausgewählt wird. Da sich die beiden Bedingungen ausschließen, tritt hier kein Konflikt auf.

Leistungsbewertung

Mit Hilfe der Leistungsbewertung sollen nun die Fragestellungen beantwortet werden, um deren willen das Modell erstellt wurde. Da die Implementierungsarbeiten zum Zeitpunkt der Fertigstellung dieser Arbeit noch nicht abgeschlossen waren, wurde das zu untersuchende spezielle Modell von Hand in ein farbloses umgewandelt. Dieses wurde dann mit Hilfe der Untersuchungskomponenten von TimeNET [33] bewertet. Da die zugrundeliegenden Analyse- und Simulationstechniken bis auf die Anpassung an die spezielle Netzklasse dieselben sind, lassen sich die Ergebnisse ohne weiteres übertragen. Modellobjekte und Leistungsparameter im Text beziehen sich der besseren Verständlichkeit halber immer auf das spezielle Petri-Netz.

Im verwendeten Beispiel treten viele feste Zeiten für Transporte und Montageaufgaben auf. Diese müssen für eine genaue Abbildung mit deterministischen Transitionen modelliert werden. Die Einschränkung der verwendeten Analysetechnik ist damit nicht mehr erfüllt, nach

der in jeder Markierung des Modells höchstens eine Transition mit nichtexponentieller Schaltzeit aktiviert sein darf. Aus diesem Grund wurden alle nachfolgenden Untersuchungen mit Hilfe der Simulation durchgeführt. Am Schluß dieses Abschnitts werden die Techniken der Simulation und Analyse anhand des Beispiels einander gegenübergestellt und der Einfluß der Wahl der Schaltzeit untersucht.

Ein Simulationsexperiment ist stets zufallsbehaftet. Trotz der eingesetzten Techniken zur Varianzschätzung treten Ungenauigkeiten auf. Diese können zu nicht ganz korrekten Kurvenverläufen führen. Alle nachfolgend beschriebenen Simulationsläufe wurden mit einem Konfidenzintervall von 90 Prozent und einem maximalen relativen Fehler von 10 Prozent durchgeführt.

Zunächst wurde das Modell mit den aus der Realität bekannten Werten der Bearbeitungszeiten, Puffergrößen usw. untersucht. Das entscheidende Leistungsmaß der Montagelinie ist ihr Durchsatz, also die pro Zeiteinheit (im folgenden pro Stunde) produzierte Anzahl an Gurtstrammern. Daraus läßt sich die zu erwartende Produktionsleistung z.B. während einer Schicht berechnen. Der Durchsatz kann an einer Transition bestimmt werden, die alle fertigen Gurtstrammer passieren müssen, wie z.B. S160 auf der Primärseite. Die Angabe des Leistungsparameters auf dieser Seite hat demnach die Form $\text{Durchsatz} = \text{TP}[\text{S160}]$. Das Ergebnis dieses ersten Simulationslaufs ist eine Produktionsleistung von 567 Teilen pro Stunde und entspricht sehr gut dem tatsächlichen Wert. Damit wurde zunächst soweit möglich der Nachweis erbracht, daß das Modell die realen Gegebenheiten abbildet.

Entsprechend der Aufgabenstellung werden nun verschiedene Einflüsse auf den Durchsatz der Montagelinie untersucht, um ein besseres Verständnis dafür zu entwickeln. Dazu gehört die Ermittlung von Engpässen, des Einflusses der Werkstückträgeranzahl und des Ausfallverhaltens.

In einer speziell für eine Fertigungsaufgabe konzipierten Montagelinie mit (innerhalb der Karrees) linearer Struktur ist eine gleichmäßige und hohe Auslastung der einzelnen Stationen

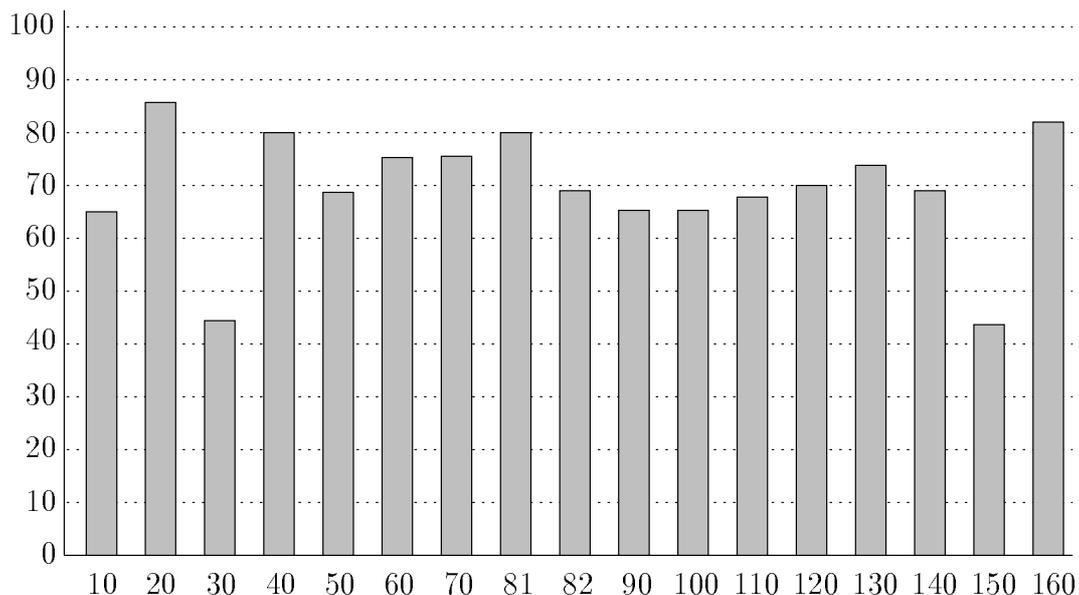


Abbildung 6.7: Auslastung der Stationen

wichtig für einen hohen Durchsatz. Es ist klar, daß das Gesamtsystem nicht schneller sein kann als die langsamste Station. Darum wurde die Auslastung der Stationen untersucht. Als Leistungsparameter wurde z.B. für die Station 20 $AuslastungS20 = P[WTV.leer \leq \theta P20]$ definiert.

Es ergibt sich das in Bild 6.7 gezeigte Balkendiagramm, in dem für jede Station die prozentuale Auslastung dargestellt ist. Die Stationen 20 und 160 sind am häufigsten aktiviert. Sie haben die höchsten auf ein Teil bezogenen Bearbeitungszeiten und stellen offensichtlich Engpässe der Montagelinie dar. Durch die vorrangige Beschickung der Station 81 ist diese deutlich stärker belastet als die identische Station 82. Die hohe Auslastung der Umsetzstation 40 ist dadurch zu erklären, daß sie mit Werkstückträgern beider angrenzender Montagekarrees beschäftigt sein kann. Besonders gering sind der Handarbeitsplatz 30 und die Prüfstation 150 ausgelastet. Dies steht in Übereinstimmung mit ihren stark unterdurchschnittlichen Bearbeitungszeiten.

Um die langsamsten Stationen erkennen zu können, müssen neben den Bearbeitungszeiten auch die Ausfälle mit betrachtet werden. Eine Maschine liefert ja nur dann den ihrer Bearbeitungszeit entsprechenden Durchsatz, wenn sie funktionstüchtig ist. Mit einem kleinen Nebenmodell wurde darum für jede Station untersucht, mit welcher Wahrscheinlichkeit sie zu einem Zeitpunkt ausgefallen ist. Dazu wurden die aus der Mitschreibung gewonnenen Daten über die Zwischenausfallzeiten, Reparaturdauern usw. entsprechend dem bereits beschriebenen Verhaltensmodell herangezogen. Die Ergebnisse zeigt die Tabelle 6.7.

Station	Bearbeitungszeit	Ausfallwahrsch.	maximaler Durchsatz	
			ohne Ausfälle	mit Ausfällen
10	4,003	0.02172	899	874
20	5,196	0.03451	693	663
30	2,819	-	1277	1277
40	3,660	0.01938	984	959
50	4,203	0.02525	856	829
60	4,423	0.05775	814	754
70	4,450	0.06515	809	743
81	3,395	0.03451	915	875
82	4,033	0.03451	892	854
83	4,380	0.11784	822	703
90	4,080	0.00927	882	872
100	3,830	0.07384	940	851
110	3,713	0.07384	969	877
120	3,900	0.03612	923	880
130	4,439	0.03764	811	772
140	4,373	-	823	823
150	2,760	-	1304	1304
160	5,195	-	693	693

Tabelle 6.7: Ausfallwahrscheinlichkeit und maximaler Durchsatz der Stationen

In ihr ist für jede Station die normale Bearbeitungszeit, die aus dem kleinen Modell bestimmte Wahrscheinlichkeit für eine defekte Maschine sowie die maximal erreichbaren Durchsätze mit und ohne Ausfälle aufgelistet. Die Bearbeitungszeiten und Durchsätze beziehen sich einheitlich auf einen Gurtstrammer. Einige Maschinen sind häufig ausgefallen, Station 83 beispielweise zu 11 Prozent. Entscheidend ist aber letztlich der theoretisch mögliche Durchsatz einer Station unter Beachtung der Ausfälle. In der Tabelle sind die niedrigsten drei Werte hervorgehoben. Station 83 beliefert die Stationen 81 und 82 mit Druckrohren. Im Verhältnis zu den anderen Stationen ist ihr Aufbau nicht besonders kompliziert. Diese demnach vergleichsweise „billige“ Maschine verzögert die restlichen Stationen darum unnötig. Im Endmontagekarree ist die Handarbeitsstation 160 am langsamsten, obwohl sie nicht durch Ausfälle verzögert wird.

Um den Einfluß von Ausfällen auf den Durchsatz des Gesamtsystem zu untersuchen, wurde ein etwas verändertes Modell verwendet. In diesem ist das Ausfallverhalten aller fehlerbehafteten Stationen identisch und so vereinfacht, daß nur eine exponentielle Ausfall- und Reparaturrate angegeben werden kann. Diese Parameter wurden im folgenden Versuch so variiert, daß jeweils eine bestimmte Intaktwahrscheinlichkeit und Dauer eines Ausfalls festgelegt ist.

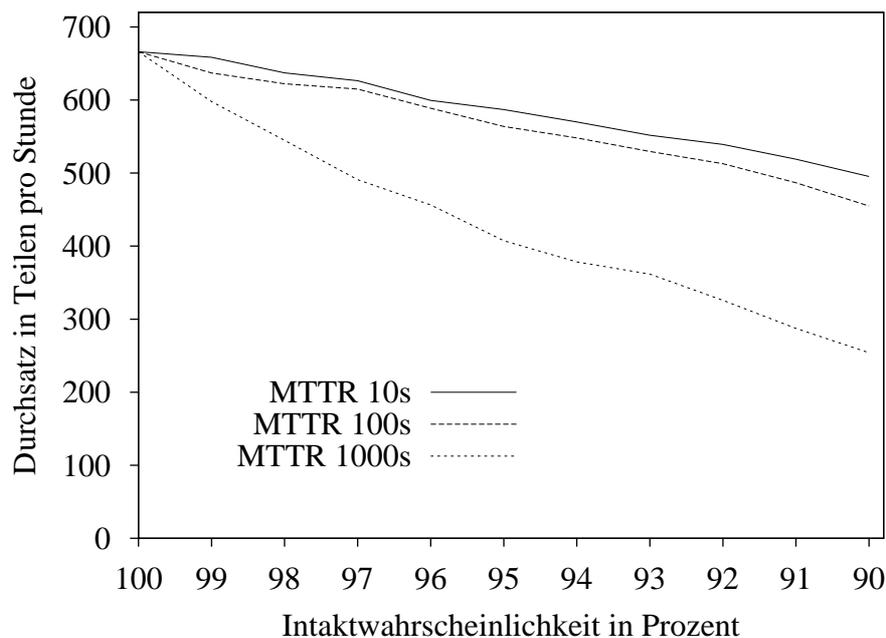


Abbildung 6.8: Durchsatz gegenüber Ausfallverhalten

Bild 6.8 zeigt die Ergebnisse dieser Simulationsläufe. Die Intaktwahrscheinlichkeit wurde von 100 bis 90 Prozent verringert, wobei der Durchsatz stark abfällt. Störungen wirken sich offensichtlich stark auf den Durchsatz aus. Interessant ist der ebenfalls große Einfluß der Ausfalldauer (MTTR). Wenn bei gleichbleibender Störungswahrscheinlichkeit die durchschnittliche Dauer einer Störung (und damit auch die Zeit zwischen zwei Störungen) erhöht wird, können die zwischen den Stationen liegenden Puffer dies schlechter ausgleichen. Der Durchsatz sinkt dann, obwohl die Stationen genauso oft ausgefallen sind.

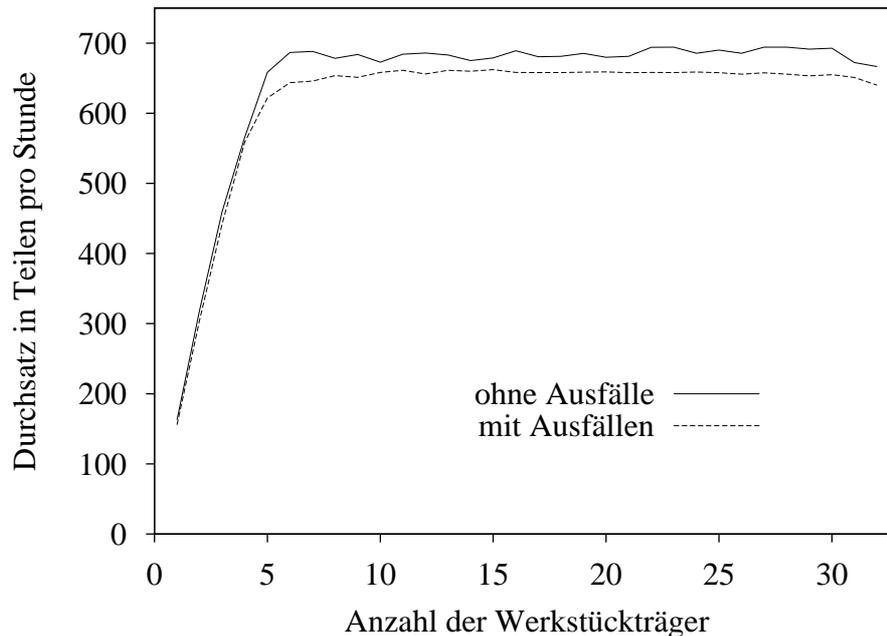


Abbildung 6.9: Durchsatz gegenüber Werkstückträgeranzahl im Vormontagekarree

Diese Wirkung der Puffer wird sicherlich größer sein, wenn die Anzahl der Pufferplätze und Werkstückträger erhöht wird. Zusätzliche Werkstückträger können außerdem problemlos nachträglich dem fertigen System hinzugefügt werden. Aus diesen Gründen wurden für die drei Montagekarrees einzeln der Einfluß der Werkstückträgeranzahl auf den Durchsatz sowohl mit als auch ohne Ausfälle untersucht.

Bild 6.9 zeigt den Durchsatz des Vormontagekarrees in Abhängigkeit von der eingesetzten Werkstückträgeranzahl, die zwischen Eins und 32 variiert wurde. Bis zu einer Anzahl von 5 erhöht sich der Durchsatz sehr stark und in etwa linear, danach kaum noch. Für dieses Modell wurde außerdem untersucht, inwieweit der Durchsatz für eine extrem hohe Anzahl Werkstückträger wieder absinkt. Dies ist offensichtlich erst dann der Fall, wenn das Montagekarree bereits fast an der Grenze seines Fassungsvermögens angelangt ist. Die tatsächlich eingesetzte Menge von 12 Werkstückträgern erscheint als guter Kompromiß zwischen ihren Anschaffungskosten und einem hohen Durchsatz. Die durch Ausfälle verursachte Minderung der Produktionsleistung ist relativ gering und liegt bei etwa fünf Prozent.

Die unter Beachtung der Ausfälle langsamste Station des Vormontagekarrees ist Station 20 (siehe Tabelle 6.7). Ihr theoretisch erreichbarer Durchsatz liegt bei 663 Werkstücken pro Stunde. Der in der Simulation für 12 Werkstückträger erzielte Wert von 655 zeigt, daß durch die Kopplung der Stationen kaum Durchsatzverluste auftreten. Der maximale Durchsatz der Station 20 wird durch Ausfälle nur um etwa 5 Prozent verringert. Eine signifikante Optimierung des Vormontagekarrees ist demnach lediglich durch eine Beschleunigung der Station 20 erreichbar.

Im Bild 6.10 ist der Durchsatz des Hauptmontagekarrees in Abhängigkeit von der eingesetzten Werkstückträgeranzahl dargestellt. Zur Erzielung des maximalen Durchsatzes sind hier mindestens 30 Werkstückträger notwendig, im gestörten System steigt der Durchsatz

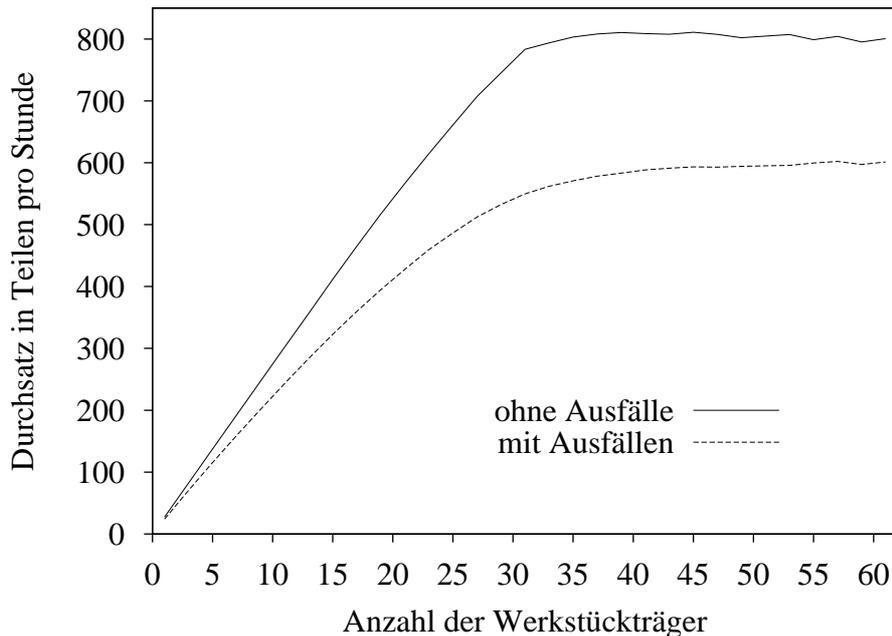


Abbildung 6.10: Durchsatz gegenüber Werkstückträgeranzahl im Hauptmontagekarree

auch darüber noch geringfügig weiter. Die verwendeten 50 Werkstückträger erscheinen etwas knapp gewählt, da in der Realität immer einige von ihnen am Reparaturplatz warten und damit dem eigentlichen Montagevorgang verloren gehen.

Der große Einfluß der Ausfälle im Hauptmontagekarree zeigt sich in einer um 35 Prozent verringerten Produktionsleistung. Engpaß ist hier Station 83, deren maximaler Durchsatz unter Berücksichtigung von Ausfällen bei 703 Werkstücken pro Stunde liegt. Im ungestörten System wäre theoretisch ein Durchsatz von 809 zu erzielen (Station 70). Da der tatsächliche Durchsatz unter 600 liegt, ist Optimierungspotential vor allem im Ausfallverhalten sowie den langsamen Stationen vorhanden. Außerdem sollten die sehr ungleichmäßig verteilten Pufferplätze ausgeglichen sowie die Werkstückträgeranzahl erhöht werden.

Die Ergebnisse der Simulationsläufe für das Endmontagekarree zeigt Bild 6.11. Der maximale Durchsatz wird ungefähr bei 15 Werkstückträgern erreicht. Auffällig ist hier, daß Ausfälle einen zu vernachlässigenden Einfluß auf den Durchsatz haben. Der Grund dafür liegt darin, daß die langsamste Station 160 eine nicht von Störungen behaftete Handarbeitsstation ist. Tatsächlicher Durchsatz (651 Werkstücke pro Stunde) und theoretisch an Station 160 erreichbarer (693) unterscheiden sich nicht sehr. Zur Verbesserung der Produktionsleistung im Endmontagekarree muß demnach die Station 160 beschleunigt werden.

Durch die Kopplung der gestörten Einzelkarrees geht der Durchsatz noch einmal auf 567 Gurtstrammer pro Stunde zurück. Dieser Wert wird offensichtlich größtenteils durch das unter Fehlereinfluß langsamste Hauptmontagekarree (Durchsatz einzeln: 594 Gurtstrammer) bestimmt. Optimierungen sollten sich aber trotzdem nicht nur auf das Hauptmontagekarree konzentrieren, da der Einzeldurchsatz der anderen Karrees nicht viel höher liegt.

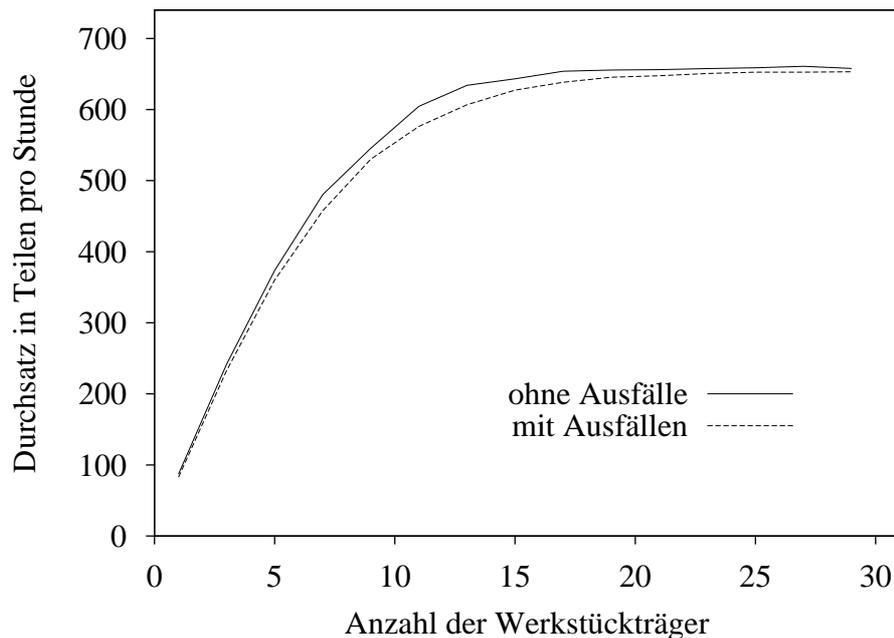


Abbildung 6.11: Durchsatz gegenüber Werkstückträgeranzahl im Endmontagekarree

Im folgenden werden aufeinander aufbauend mehrere Vorschläge zur Erhöhung der Leistungsfähigkeit der Anlage gemacht und ihre Auswirkungen anhand des Modells überprüft.

Insbesondere im Hauptmontagekarree sind die vorhandenen Pufferplätze sehr ungleichmäßig verteilt. Eine Angleichung der Puffergrößen könnte zu einem besseren Ausgleich von Ausfällen führen, da dann durch eine gestörte Station die folgende nicht so schnell blockiert wird. Dabei bleiben die Fahrzeiten pro Pufferplatz und die Gesamtanzahl der Plätze unverändert. Außerdem wird die Anzahl der eingesetzten Werkstückträger etwas erhöht, so daß im Vor-, Haupt- und Endmontagekarree 20, 60 bzw. 40 von ihnen vorhanden sind. Im Ergebnis erhöht sich der Durchsatz der Gesamtanlage von 567 auf 590 Werkstücke pro Stunde.

Die Untersuchungen zeigten, daß die bei der Planung der Montagelinie kaum beachteten Ausfälle einen großen Einfluß auf ihre Leistungsfähigkeit haben. Durch eine bessere Wartung und den Austausch besonders verschleißgefährdeter Teile in den Stationen gegen widerstandsfähigere könnte die mittlere Betriebszeit bis zu einem Ausfall erhöht werden, angenommen auf mindestens 2000 Sekunden. Langwierige Reparaturen sind durch effizienter arbeitende Monteure sowie Bereitstellung der am häufigsten benötigten Ersatzteile an den Stationen verkürzbar. Es sei angenommen, daß durch entsprechende Maßnahmen die Dauer der langen Ausfälle für alle Stationen um ein Drittel verkürzt werden kann. Der Durchsatz eines so veränderten Systems liegt dann bei 599 Werkstücken, erhöht sich also kaum im Vergleich zum Vormodell. Allerdings ist auch völlig ohne Ausfälle nur ein Durchsatz von 647 erreichbar, was bereits recht nahe am erzielten Wert liegt.

Weitere Produktivitätssteigerungen sind demnach nur durch Beschleunigen der Stationen erzielbar. Dazu können zunächst die eingesetzten Arbeiter gleichmäßiger auf die Handarbeitsstationen aufgeteilt werden, so daß ohne zusätzliche Arbeitskräfte dieselbe Bearbeitungszeit an jeder Station benötigt wird. Außerdem sei angenommen, daß alle bisher langsameren Sta-

tionen so beschleunigt werden, daß sie den ursprünglich geplanten Bearbeitungstakt von vier Sekunden erreichen. Ihr tatsächlicher Durchsatz wird dabei weiterhin durch Ausfälle verringert. Das entstehende Modell hat einen Durchsatz von 693 Teilen pro Stunde und erreicht damit eine um 22 Prozent erhöhte Produktionsleistung als das Ausgangsmodell. Allerdings werden so erst 77 Prozent der theoretisch möglichen Leistung von 900 Werkstücken erzielt, die durch die Taktzeit von vier Sekunden vorgegeben ist.

Nach den vorangegangenen Untersuchungen des Anwendungsbeispiels werden nun die Techniken der numerischen Analyse und Simulation einander gegenübergestellt. Dies geschieht anhand eines vereinfachten Modells des Vormontagekarrees der beschriebenen Montageanlage, in der die Ausfälle der Stationen außer acht gelassen wurden. Für die Anwendbarkeit der numerischen Analysetechnik darf maximal eine Transition mit nichtexponentieller Schaltzeit pro Markierung aktiviert sein. Zum Vergleich zwischen Simulation und Analyse wurde darum zusätzlich ein abgewandeltes Modell untersucht, in dem alle Transitionen eine exponentielle Schaltzeit bei sonst gleichen Eigenschaften haben.

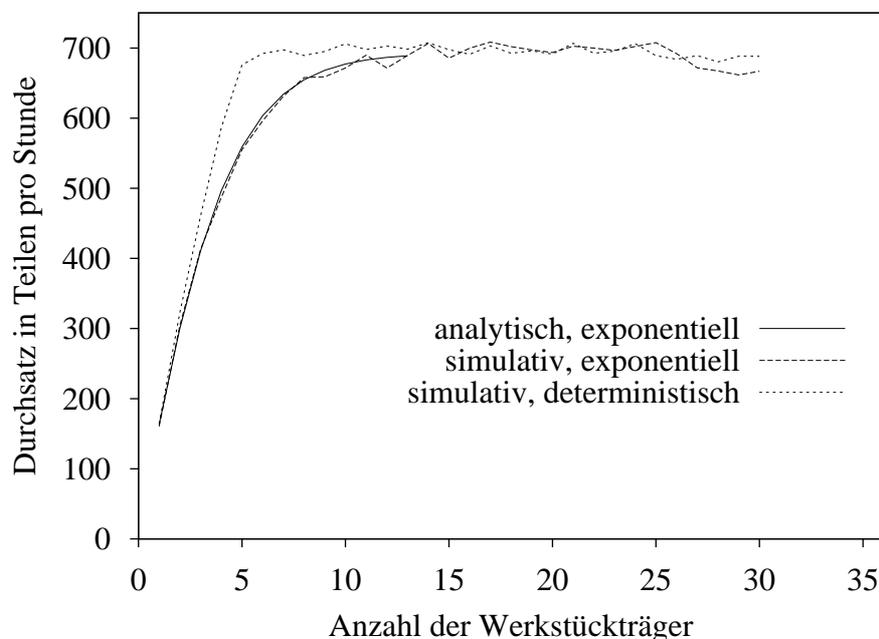


Abbildung 6.12: Gegenüberstellung von Simulation und Analyse

Im Bild 6.12 sind die Ergebnisse der Berechnung des Durchsatzes des Vormontagekarrees dargestellt. Im Vergleich zu den analytisch gewonnenen (exakten) Ergebnissen zeigen die simulativ bestimmten Kurven Abweichungen, die sich in welligen Kurvenverläufen niederschlagen. Die durch Simulation und Analyse berechneten Kurven für das Modell mit exponentiellen Schaltzeiten stimmen aber wertmäßig gut überein. Die Genauigkeit der Simulationsergebnisse ist demnach ausreichend.

Der Einfluß der Wahl der Schaltzeit ist für hohe Durchsätze gering, im wichtigen Bereich bis zu 15 Werkstückträgern jedoch nicht zu vernachlässigen. Es zeigt sich, daß für eine Bestimmung der mindestens erforderlichen Werkstückträgeranzahl unbedingt die realen deterministischen Verteilungen verwendet werden müssen. Aus diesem Grund kann hier nur die Simulation eingesetzt werden.

Anzahl Werkstückträger	andauernde Markierungen	Berechnungsdauer in Sekunden		
		Analyse exponentiell	Simulation exponentiell	Simulation deterministisch
1	9	3	7	6
2	39	3	6	7
3	119	3	6	7
4	294	3	6	7
5	630	5	7	6
6	1218	17	7	6
7	2178	87	6	6
8	3663	392	7	6
9	5863	1673	7	6
10	9009	5320	8	7
11	13372	15216	6	7
12	19218	37783	6	7
13	26758	90150	6	6
14			7	6
15			7	7
16			6	7

Tabelle 6.8: Berechnungszeiten von Analyse und Simulation

Die Kurve für die numerische Analyse wurde in Bild 6.12 nur bis zu 13 Werkstückträgern berechnet, da der zeitliche Aufwand der Analysen sehr groß wurde. Tabelle 6.3 zeigt die Anzahl der andauernden Zustände des Modells sowie die für die Berechnung notwendigen Zeiten in Abhängigkeit von der gewählten Methode und der Werkstückträgeranzahl. Die Messungen wurden auf einer DEC Alpha 3000 Workstation durchgeführt.

Bei der numerischen Analyse wird zunächst der reduzierte Erreichbarkeitsgraph aufgebaut und danach das sich ergebende lineare Gleichungssystem gelöst (siehe Kapitel 7), da es sich hier nur um ein GSPN handelt. In diesem speziellen Fall benötigte die Berechnung des ja noch relativ kleinen Erreichbarkeitsgraphen für 26758 Zustände lediglich 13 Sekunden, während die Lösung des Gleichungssystems bereits über einen Tag benötigte. Dies ist allerdings kein typisches Beispiel; für andere Modelle sind in dieser Zeit Lösungen für sehr viel größere Zustandsräume berechenbar. Es zeigt aber auch das Problem der numerischen Analyse auf, die für die implementierten Techniken stark von der Größe des Zustandsraums abhängt und hier bereits für ein sehr kleines Modell sehr lange Ausführungszeiten benötigt. Der zur Verfügung stehende Hauptspeicher war hier kein Problem, kann aber bei größeren Modellen zur Einschränkung werden.

Im Vergleich dazu benötigen die Simulationsexperimente für beide Modelle im gesamten betrachteten Bereich konstant ungefähr 7 Sekunden, um die geforderte Genauigkeit zu erreichen. Mit einer numerischen Analyse hat man im Gegensatz dazu die Gewißheit, daß das Ergebnis exakt ist. Dies wird aber mit einer für das Beispiel sehr viel längeren und von Modellparametern abhängenden Berechnungszeit erkaufte. Die Laufzeit eines Simulationsexperimentes hängt weniger von der Größe des Erreichbarkeitsgraphen des Modells ab als von

der Art der Ergebnisparameter. Werden diese stark durch seltene Ereignisse beeinflusst, sind lange Simulationszeiten die Folge. In unserem Beispiel wurde nur der Durchsatz untersucht, eine offensichtlich für die Simulation unkritische Größe. Bei der Leistungsbewertung eines Modells kann also abhängig von der Art des Modells und den Fragestellungen jeweils eine der beiden zur Verfügung stehenden Techniken eingesetzt werden. Dabei muß anhand der Genauigkeit, der Modellierungsmächtigkeit und der Ausführungszeiten abgewogen werden.

6.4 Fazit

In diesem Kapitel wurden die in der Dissertation entwickelten Modellierungs- und Untersuchungsmethoden auf ein reales Beispiel angewendet. Es zeigte sich, daß die Abbildung des Fertigungsprozesses mit den speziellen farbigen Petri-Netzen einfach und übersichtlich möglich war. Die entstehenden Modelle der Struktur und der Arbeitspläne ähneln in ihrem Aufbau dem Layout bzw. gebräuchlichen grafischen Beschreibungsmitteln und sind dadurch gut verständlich. Durch die Verwendung von Teilmodellen aus einer Bibliothek wird der Modellierungsprozeß schneller und weniger fehleranfällig. Bei der Erstellung der Arbeitsplanmodelle könnte eine Unterstützung durch das Softwarewerkzeug die Arbeit weiter erleichtern, indem nur zwischen den aktuell möglichen Netzelementen ausgewählt werden muß. Die angebbaren Leistungsmaße waren für die zu beantwortenden Fragestellungen geeignet und ausreichend. Die Fähigkeit der Modellierungsmethode, verschiedene Arbeitspläne für unterschiedliche Werkstücke innerhalb eines Fertigungssystems zu verarbeiten, wurde für das Anwendungsbeispiel nicht benötigt.

Ein großes Problem bei der Modellierung besteht in der Auswahl und Gewinnung der relevanten Daten für Bearbeitungszeiten usw., ist aber unabhängig von dem gewählten Beschreibungsmittel. Problematisch ist außerdem die Modellierung von Werkstücken, die sich gemeinsam auf einem Werkstückträger, einer Palette o.ä. befinden. Solange sich die zusammengefaßten Werkstücke nicht unterscheiden, kann — wie im Beispiel getan — leicht davon abstrahiert werden. Wenn es sich jedoch um unterschiedliche Produkte oder solche mit verschiedenen Bearbeitungsstufen handelt, ist dies nicht mehr so einfach möglich. Ein Ausweg kann dann die Definition von verschiedenen Zuständen der Werkstückträger für alle möglichen Kombinationen der transportierten Produkte sein. Anhand dessen können dann entsprechende Arbeitspläne modelliert werden. Für das Anwendungsbeispiel wurde dies in [24] zur Abbildung der Fehlerzustände der Werkstücke durchgeführt. Infolgedessen können die Arbeitsplanmodelle sehr umfangreich werden. In diesem Zusammenhang wären zusätzliche Marken zur Modellierung von Behältern denkbar, die andere Produktmarken enthalten dürfen.

Im Anschluß an die Modellierung des Anwendungsbeispiels wurden die Untersuchungsverfahren eingesetzt. Die strukturelle Analyse war besonders während der Modellerstellung hilfreich, um grundlegende Eigenschaften der entstehenden speziellen Petri-Netze zu überprüfen. Spezifikationsfehler können so auch in großen Modellen gefunden werden. Da eine derartige Untersuchung sehr schnell abläuft, werden viele Fehler nicht erst nach Ablauf eines längeren Simulationsexperiments entdeckt.

Aufgrund der für Modelle dieser Größe beschränkten Anwendbarkeit der numerischen Analyseverfahren wurde mit Hilfe von Simulationsexperimenten eine Leistungsbewertung des Modells durchgeführt. Dabei wurde die Leistungsfähigkeit des Systems unter Beachtung des Ausfallverhaltens (*performability*) untersucht. Die gute Übereinstimmung des Ergebnisses mit dem realen Wert weist auf ein ausreichend genaues Modell und die zuverlässige Funktion der Untersuchungsmethoden hin. Wären diese Untersuchungen bereits in der Planungsphase der Montageanlage durchgeführt worden, hätte die Produktionsleistung vorausgesagt und Schritte zu ihrer Verbesserung ergriffen werden können. Die weiteren Experimente zeigen die Größe des Einflusses verschiedener Faktoren auf die Leistungsfähigkeit der Anlage auf. Es wurden Vorschläge zur Erhöhung des Durchsatzes gemacht und diese anhand des Modells überprüft. Leider sind sie an der laufenden Anlage nachträglich kaum noch zu realisieren. Dies zeigt die Notwendigkeit der Leistungsbewertung zu einem frühen Zeitpunkt der Fertigungssystemplanung.

Zusammengefaßt erwiesen sich die entwickelten Modellierungs- und Untersuchungsmethoden als für das Beispiel geeignet und tauglich zur Beantwortung der relevanten Fragestellungen.

Kapitel 7

Ein Softwarewerkzeug zur Modellierung und Bewertung von Fertigungssystemen mit speziellen Petri-Netzen

In diesem Kapitel wird ein Softwarewerkzeug beschrieben, in dem die in der vorliegenden Dissertation entwickelten Methoden realisiert wurden. Es ist als Erweiterung des Modellierungs- und Analysewerkzeugs TimeNET [33, 34] implementiert. Mit TimeNET konnten bisher farblose zeitbehaftete Petri-Netze grafisch eingegeben, strukturell untersucht, analysiert und simulativ bewertet werden.

Aufgrund der großen Unterschiede in den behandelten Modellklassen mußte für die im Kapitel 2 vorgestellte Art farbiger Petri-Netze eine neue Benutzungsoberfläche entwickelt werden, die im folgenden Abschnitt 7.1 beschrieben wird. Im darauffolgenden Teil werden die Analyse- und Simulationskomponenten erklärt und der Ablauf der Bewertungsverfahren geschildert. Den Abschluß des Kapitels bildet eine Zusammenfassung.

Es gibt eine Vielzahl von Werkzeugen zur Modellierung und Untersuchung mit Petri-Netzen. GreatSPN [15] kann GSPNs und farbige *stochastic well-formed nets* analysieren. DSPNexpress [50] ist als Vorläufer von TimeNET ebenso wie dieses speziell für Petri-Netze mit nichtexponentiellen Schaltzeitverteilungen entwickelt worden. UltraSAN [22] ist ein Werkzeug für die Untersuchung von *stochastic activity networks*, die mit GSPNs vergleichbar sind. Die Modellierung mit farbigen Petri-Netzen unterstützt DesignCPN [40], jedoch nicht ihre numerische Analyse.

Neben den genannten allgemeinen Werkzeugen auf der Basis von Petri-Netzen existieren auch solche, die auf Fertigungssysteme spezialisiert sind. Für die Modellierung, strukturelle Analyse und interaktive Simulation von Fertigungssystemen wurde in [14] das Softwarewerkzeug XPN-FMS vorgestellt. Für die Untersuchung ist die Netzklasse aber auf *decision-free Petri nets* eingeschränkt. Das Programmsystem GRAMAN [75] wurde bereits im Abschnitt 2.4 beschrieben.

7.1 Eine generische Benutzungsoberfläche

Für die Arbeit mit der vorgestellten Modellierungsmethode ist eine grafische Benutzungsoberfläche notwendig. Da die entwickelten Methoden innerhalb des Modellierungs- und Analysewerkzeugs TimeNET realisiert werden, lag die Verwendung der dafür bereits existierenden Benutzungsoberfläche nahe. Diese unterstützt jedoch lediglich Petri-Netze ohne unterscheidbare Marken und Hierarchien. Erfahrungsgemäß treten bei der Implementierung neuer Funktionen oder anderer Modellelemente innerhalb eines vorhandenen Softwarewerkzeugs große Schwierigkeiten auf, wenn dieses nicht von Anfang an erweiterbar konzipiert wurde. Darum wurde entschieden, für die Realisierung der hier vorgestellten Methoden eine neue Benutzungsoberfläche zu implementieren. Dieser Abschnitt beschreibt die Konzeption dieses Programmsystems sowie seine Bedienung. Eine genauere Darstellung der Implementierung enthält [43].

Innerhalb der Methode werden verschiedene grafische Beschreibungsmethoden für Fertigungssysteme eingesetzt. Ein benutzungsfreundliches Softwarewerkzeug muß den Umgang mit allen diesen Modellarten unterstützen. Darüber hinaus ist durch nicht vorhersehbare Weiterentwicklungen eine leichte Änderbarkeit der unterstützten Netzarten sowie Funktionen notwendig. Aus diesen Gründen wurde eine generische Benutzungsoberfläche für hierarchische grafische Modelle realisiert. Jeder Modelltyp wird als eine *Netzklasse* bezeichnet.

Allen verwendeten Beschreibungsmitteln ist gemeinsam, daß ein Modell aus *Seiten* bestehen kann, die die eigentlichen Modellelemente enthalten. Modellelemente sind entweder *Knoten* eines Graphen oder gerichtete *Kanten*, die diese Objekte verbinden. Kanten dürfen möglicherweise nur bestimmte Knoten miteinander verbinden. Knoten und Kanten haben *Attribute* wie Name, Schaltzeit, Markierung oder Gewicht. Es gibt in jeder Netzart Typen von Kanten und Knoten, beispielsweise Stellen, Transitionen und Verbindungskanten. Die Netzelemente eines Typs besitzen dieselben Attribute, denen aber verschiedene Werte zugewiesen sein können. Außerdem ist ihre grafische Darstellung gleich, wenn sie nicht durch bestimmte Attributwerte abgeändert wird. Einem Knoten kann eine Unterseite zugeordnet sein, mit der er verfeinert wird. In dieser Unterseite sind Verweise auf die Knoten eingebettet, die auf der Oberseite mit dem Knoten verbunden sind.

Die Definition einer Netzklasse soll ohne Programmieraufwand und Kenntnisse der Implementierung der Benutzungsoberfläche möglich sein. Sie erfolgt darum in einer Textdatei, deren Syntax im Anhang D beschrieben ist. In einer solchen Netzklassendatei werden neben dem Namen der Netzklasse die auftretenden Kanten- und Knotentypen beschrieben. Für jedes dieser Netzelemente werden grafische Darstellung und Attribute angegeben. Die Attributtypen sind ähnlich wie in einer Programmiersprache aufgebaut. Es gibt die Grundtypen natürliche Zahl, reelle Zahl, Aufzählungstyp, boolescher Typ, Bezeichner und mehrzeiligen Text. Aus diesen Bausteinen können bei Bedarf zusammengesetzte Attribute gebildet werden. Dies geschieht mit einer Struktur oder Liste, die andere Attribute zusammenfassen. Für die Grundattribute sind zusätzliche Informationen angebbbar, z.B. minimaler und maximaler Wert einer Zahl, die Ausprägungen des Aufzählungstyps und andere. Außerdem werden die für ein Knotenobjekt gültigen Verbindungsregeln spezifiziert, und ob das Objekt hierarchisch verfeinerbar ist. Normalerweise ist die Unterseite eines Verfeinerungsknotens vom selben Netztyp wie die Oberseite. Es gibt aber auch die Möglichkeit, als Typ der Unterseite eine andere Netzklasse zu referenzieren. Dies erlaubt z.B. die Verwaltung von Struktur- und

Arbeitsplanmodellen und der Bibliotheksmodule innerhalb eines Netzes.

Die grafische Darstellung eines Knotenobjekts ist aus Grundelementen zusammengesetzt, die ebenfalls in der Netzklassendatei angegeben sind. Dazu gehören Linienzüge, Ellipsen und Ellipsenausschnitte sowie Texte, die zu Gruppen zusammengestellt werden können. Mit dem jeweiligen grafischen Kontext sind Liniendicke, Farbe, Füllzustand, Linienstil und Zeichensatz einstellbar. Die attributabhängige Darstellung eines Objektes ist ebenfalls möglich. Dazu können boolesche Ausdrücke für grafische Elemente in der Netzklassendefinition angegeben werden. Diese hängen von den Objektattributen ab und entscheiden mit ihrem Wahrheitswert darüber, ob das zugehörige grafische Element angezeigt wird oder nicht. Der einfache Grundaufbau der grafischen Beschreibung mit vektororientierten Elementen erleichtert das Skalieren, Rotieren sowie Exportieren in andere Formate wie *PostScript* oder *xfig*.

Die Netzklassendateien befinden sich in einem dafür vorgesehenen Verzeichnis, das beim Start der Benutzungsoberfläche durchsucht wird. Alternativ kann eine Netzklasse aber auch in die Oberfläche integriert oder von einem Erweiterungsmodul generiert werden. Dies geschieht beispielsweise für die im Kapitel 5 beschriebenen symbolischen Modelle, da die entsprechende Netzklasse erst zur Laufzeit aus den verfügbaren Bibliotheksmodulen erzeugt werden kann. Ein vom Benutzer eingegebenes Modell ist dann stets eine Ausprägung einer Netzklasse und wird in einer eigenen Datei gespeichert. Die Syntax dieser Datei beschreibt Anhang D.

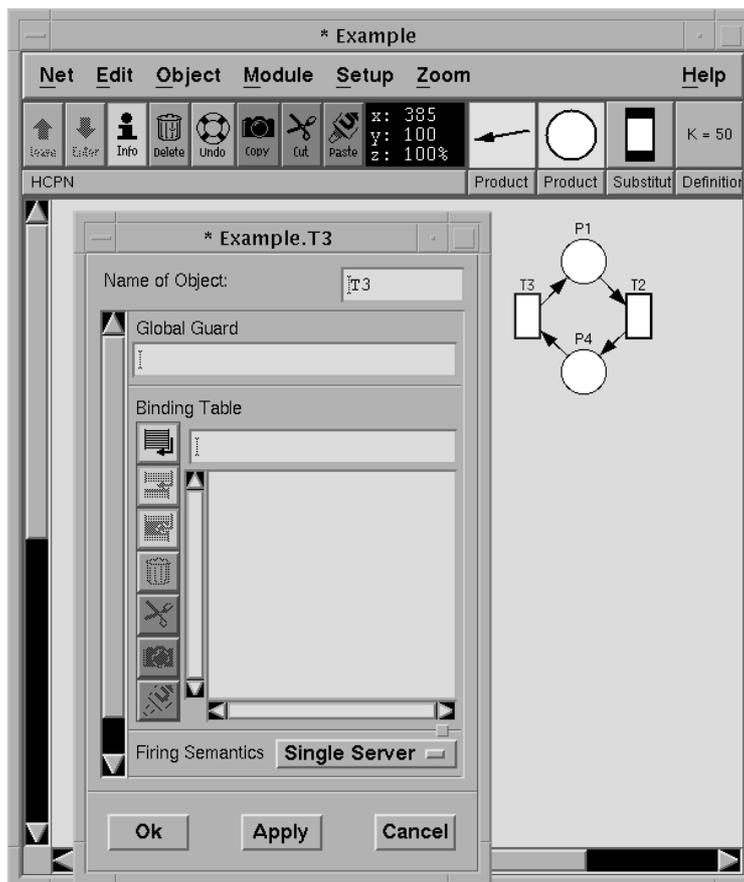


Abbildung 7.1: Bild der Benutzungsoberfläche

Das Bild 7.1 [43] zeigt eine beispielhafte Bildschirmansicht der Benutzungsoberfläche. Über die Menüleiste an ihrem oberen Rand können allgemeine Funktionen zum Laden und Speichern eines Modells, Kopieren und Einfügen von Objekten, Rückgängigmachen des letzten Befehls und ähnliches ausgeführt werden. Anzeige- und Bedienoptionen werden mit *Setup* und *Zoom* eingestellt. Darunter sind Bedienknöpfe mit Symbolen für häufig verwendete Funktionen vorhanden. Rechts davon werden die Objekte der aktuellen Netzklasse angezeigt, wobei in einem Feld zwischen Objekten aus einer Gruppe (z.B. Elementarstellen, Produktstellen) umgeschaltet werden kann. Die Belegung und Darstellung dieser Knöpfe wird durch die Benutzungsoberfläche zur Laufzeit aus der entsprechenden Netzklassendefinition erzeugt. Der große untere Bildschirmteil zeigt einen Ausschnitt des aktiven Modells, in dem es mit den in grafischen Werkzeugen üblichen Mausfunktionen direkt bearbeiten werden kann.

Die grafische Darstellung der Netzelemente innerhalb des Bearbeitungsfensters erfolgt für jedes Objekt so, wie es in der Netzklasse definiert wurde. Komplizierter zu realisieren ist die Eingabe von Attributen der Objekte, die ebenfalls textuell in der Netzklassendatei festgelegt sind. Nach einem Doppelklick auf das gewünschte Objekt erscheint ein Eingabefenster, in dem für jedes Attribut des Objekts der Attributname und ein Eingabefeld mit dem aktuellen Wert enthalten ist. Für jedes Grundattribut gibt es dafür eine spezielle Darstellungsform. Ein boolesches Attribut wird als Ein/Aus-Knopf, ein Aufzählungstyp als *popup*-Menü und Texte sowie Zahlen mit einem Eingabefeld dargestellt. Zu Listen zusammengefaßte Attribute können beliebig oft auftreten. Dies wird mit einem eingebetteten Fenster erreicht, innerhalb dessen die Listenelemente eingegeben werden können. Außerdem sind Bedienknöpfe zum Einfügen, Kopieren usw. neuer Listenelemente vorhanden. Im Bild 7.1 ist das Attributfenster der Transition T3 geöffnet. Die Attribute dieser Transition sind der Name, eine globale Schaltbedingung (Text), eine Liste von Übergängen (Liste von Texten) und die Schaltsemantik (Aufzählungstyp). Es ist möglich, ein weiteres Fenster der Benutzungsoberfläche auf einem anderen Arbeitsplatzrechner zu öffnen. Dann können verschiedene Modellierer gemeinsam an einem Modell arbeiten.

Mit der Benutzungsoberfläche können Modelle eingegeben oder verändert werden, die in einer Netzklasse definiert wurden. Eine Benutzungsoberfläche soll aber neben dem Eingeben des Modells auch zum Aufrufen der Analysealgorithmen dienen, damit der Modellierer nur eine einheitliche Umgebung benötigt. Aufgrund der Unterschiede der vielen möglichen Netzklassen muß es dafür spezielle Werkzeuge geben, für die eine Parametrisierung über eine Textdatei nicht ausreicht. Zusätzliche Programme sollten demnach einfach in die bestehende Benutzungsoberfläche einzubinden sein. Diese Programme benötigen den Zugriff auf die Daten der Oberfläche und sind jeweils nur für bestimmte Netzklassen anwendbar. Diese Anforderungen wurden im vorliegenden Softwarewerkzeug durch eine Modulschnittstelle realisiert.

Ein Modul ist dabei ein Programmteil zur Erweiterung der Funktionalität der Oberfläche, das mit ihr zusammen übersetzt und gebunden wird. Wenn ein neues Modell geladen oder erzeugt wird, fragt die Benutzungsoberfläche über eine entsprechende Funktion des Moduls nach, ob es für Modelle dieser Netzklasse aktiv ist. Das Modul kann dann z.B. einen Menüeintrag erzeugen, dessen Aufruf später zur Ausführung einer Modulfunktion führt. Für komplexere Funktionen kann ein Modul Benutzereingaben und andere Ereignisse abfangen, selbst interpretieren und Funktionen der Oberfläche aufrufen.

Ein bereits realisiertes Modul, welches ohne Einschränkung der Netzklasse arbeitet, ermöglicht den Export der grafischen Darstellung eines Modells in das Zeichenprogramm *xfig*. Auf diese Weise wurden die in dieser Arbeit gezeigten Bilder von Modellen erzeugt. Ein weiteres Modul ist für die Erzeugung des Gesamtmodells aus den getrennten Struktur- und Arbeitsplanmodellen verantwortlich. Die Modellierungstechnik mit Funktionssymbolen wird ebenfalls durch ein Modul realisiert. Es bestimmt bei seiner Aktivierung die verwendbaren symbolischen Elemente aus der Bibliothek vordefinierter Teilmodelle. Dementsprechend wird intern die Netzklasse der symbolischen Modelle generiert, die der Benutzer verwenden kann. Das Modul stellt weiterhin Funktionen zur Umwandlung des symbolischen Modells in ein spezielles Petri-Netz zur Verfügung und startet die Untersuchung des entstehenden Modells. Die Ergebnisse der Leistungsbewertung werden dann in das symbolische Modell übertragen.

Die beschriebene Benutzungsoberfläche hat sich bei der Anwendung als mächtiges Werkzeug zur benutzungsfreundlichen Eingabe grafischer Modelle erwiesen. Ändern und Erzeugen neuer Netzklassen ist leicht möglich, und über die Modulschnittstelle kann eine Vielzahl zusätzlicher Funktionen realisiert werden.

7.2 Werkzeugkomponenten zur Modellbewertung

Zur Bewertung der mit der beschriebenen Benutzungsoberfläche erzeugten Modelle müssen die in Kapitel 4 beschriebenen Techniken als Programm realisiert werden. Dabei konnten einige der bereits in TimeNET vorhandenen Komponenten verwendet bzw. abgewandelt werden. Dieser Abschnitt erläutert, wie die verschiedenen Programmteile und die Benutzungsoberfläche bei der Untersuchung eines Fertigungssystems zusammenspielen. Die Aufgaben der Analysekomponenten werden kurz erklärt.

Für den Start der numerischen Analyse und der Simulation ist jeweils ein Modul in die Benutzungsoberfläche integriert. Diese beiden Module erzeugen einen Menüeintrag, über den die Untersuchung durch den Benutzer gestartet werden kann. Sie schreiben die Modellinformation für ein angenommenes Modell mit dem Namen `net` in eine Datei `net.TABLE`. Dies hat den Vorteil, daß Untersuchungen an einem fertigen Modell auch unabhängig von der Oberfläche durchgeführt werden können.

Bild 7.2 zeigt die Anbindung der Oberfläche an die nachfolgenden Programmkomponenten und den Ablauf der Analyse- und Simulationswerkzeuge. Ellipsen enthalten Daten, die entweder intern in Prozessen existieren oder über eine Datei bzw. *Unix-socket* ausgetauscht werden. Wenn es sich um eine Datei handelt, ist der Name angegeben. Die Richtung des Datenaustauschs wird mit Pfeilen verdeutlicht. Fett gezeichnete Rechtecke mit abgerundeten Kanten symbolisieren größere Programmfunktionen. Diese werden innerhalb von Prozessen ausgeführt, die als gepunktete Rechtecke dargestellt sind. Hintereinanderliegende Objekte weisen auf mehrere ähnliche Prozesse mit gleicher Funktionalität hin.

Zunächst wird der Ablauf der numerischen Analyse geschildert, die im Bild 7.2 links dargestellt ist. Nach dem Abspeichern der Modellinformationen wird durch das Analyse-Modul der Prozeß `RRG` gestartet. Dieser liest zunächst die Datei ein und analysiert sie mit Hilfe der Unix-Werkzeuge *lex* und *yacc*, um die internen Datenstrukturen aufzubauen. Anschließend wird der reduzierte Erreichbarkeitsgraph des speziellen Petri-Netzes berechnet. Dieser enthält

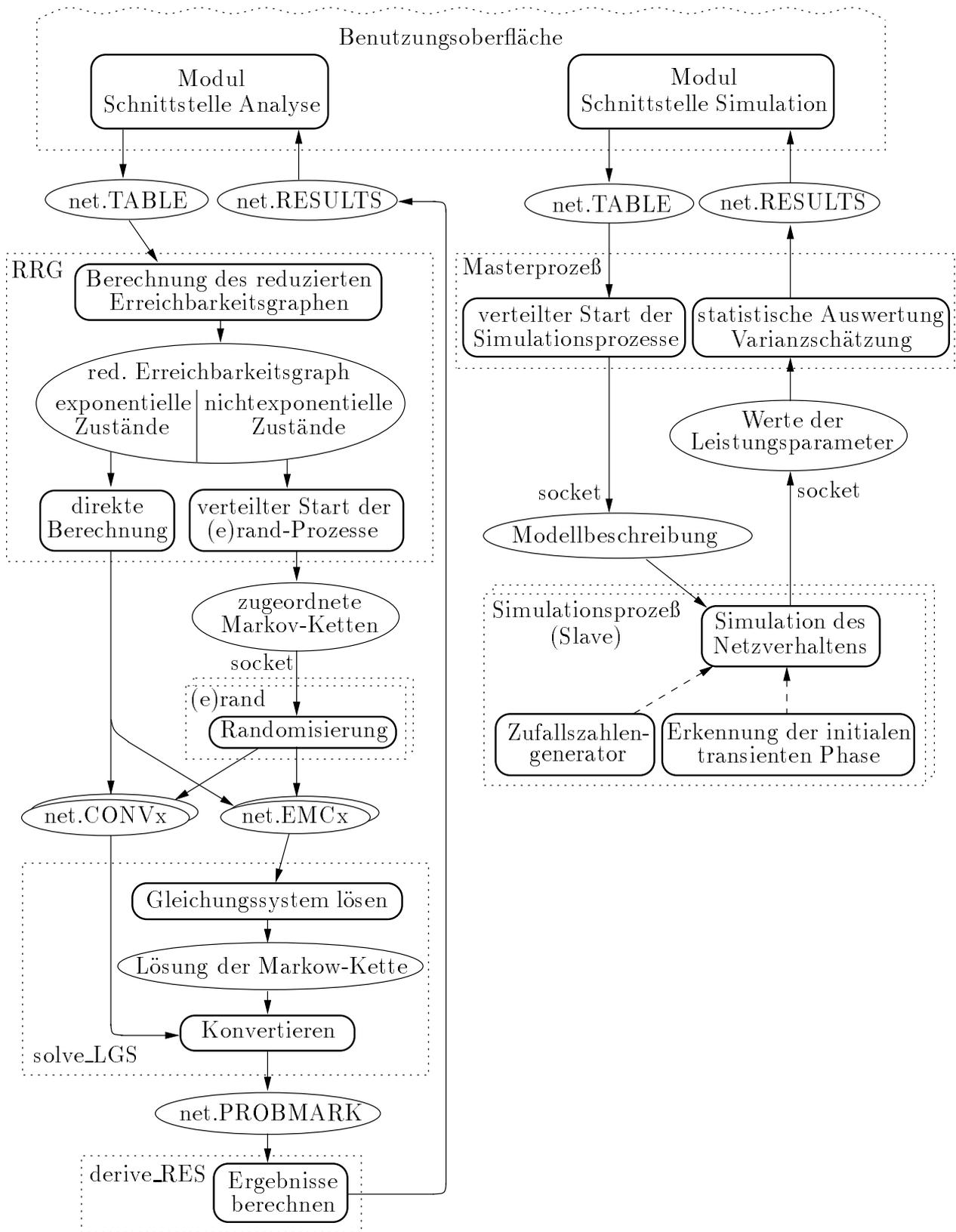


Abbildung 7.2: Ablauf der Analyse- und Simulationskomponenten

Zustände, in denen nur Übergänge mit exponentieller Schaltzeitverteilung schaltfähig sind (exponentielle Zustände). Für diese können die Übergangswahrscheinlichkeiten und mittleren Verweildauern direkt berechnet werden. Die Ergebnisse werden in die Dateien `net.EMC00` und `net.CONV00` geschrieben.

Für die Teile des Erreichbarkeitsgraphen, in denen Übergänge mit nichtexponentiell verteilter Schaltzeit aktiviert sind, muß nun für jeden Übergang ein `rand`- oder `erand`-Prozeß gestartet werden, je nachdem, ob es sich um eine deterministische oder komplexere Schaltzeit handelt. An jeden der Prozesse werden über ein `socket` die Daten der zugeordneten Markow-Kette des Übergangs gesendet. Innerhalb dieser Prozesse werden mit der Randomisierungstechnik die Übergangswahrscheinlichkeiten und Verweildauern der zugeordneten Markow-Ketten berechnet. Diese Algorithmen konnten von TimeNET übernommen werden. Prozeß i schreibt seine Ergebnisse in die Dateien `net.EMCi` und `net.CONVi`. Aufgrund der Unabhängigkeit der einzelnen zugeordneten Markow-Ketten können diese Prozesse innerhalb einer Gruppe von Arbeitsplatzrechnern verteilt ablaufen. Wenn alle Randomisierungsprozesse beendet sind, enthalten die `net.EMC`-Dateien die Matrix \mathcal{P} der Übergangswahrscheinlichkeiten der eingebetteten Markow-Kette und `net.CONV` die Matrix \mathcal{C} der Konversionsfaktoren.

Anschließend wird der ebenfalls aus TimeNET übernommene `solve_LGS`-Prozeß gestartet, der die vorher erzeugten Dateien einliest. Dann wird die Lösung des sich aus \mathcal{P} ergebenden linearen homogenen Gleichungssystems mit Hilfe eines iterativen Verfahrens (*successive overrelaxation*) oder bei dessen Fehlschlagen direkt mit der Gaußschen Eliminationsmethode berechnet. Dabei wird gleich die Bedingung mit beachtet, daß die Summe der Elemente des Ergebnisvektors Eins sein muß. Die sich daraus ergebende Lösung der eingebetteten Markow-Kette muß nun noch durch Multiplikation mit der Matrix \mathcal{C} der Konversionsfaktoren in den Zeitbereich übertragen werden. Das Ergebnis ist der Vektor π der Zustandswahrscheinlichkeiten im eingeschwungenen Zustand, der in der Datei `net.PROBMARK` gespeichert wird.

Diese Datei wird vom Prozeß `derive_RES` benötigt, der anhand der Definitionen der Leistungsparameter deren Werte berechnet und in `net.RESULTS` ausgibt. Die Ergebnisse werden dann durch das Modul der Benutzungsoberfläche gelesen, die sie als Werte der Leistungsparameter einträgt und so für den Modellierer verfügbar macht.

Ebenso wie für die Analyse gibt es ein Modul der Benutzungsoberfläche, das die Schnittstelle zu den Simulationskomponenten realisiert. Deren Ablauf ist im Bild 7.2 auf der rechten Seite abgebildet. Nach dem Schreiben der Modellinformationen in die Datei `net.TABLE` wird der sogenannte Masterprozeß gestartet. Dieser Prozeß ist für den verteilten Start und die Verwaltung der eigentlichen Simulationsprozesse verantwortlich. Jedem Simulationsprozeß (*slave*) werden das gesamte Modell und die Leistungsparameter übergeben. Dieser Prozeß führt die Simulation des Netzverhaltens durch, wobei die üblichen Schritte (Berechnen der schaltfähigen Transitionen und zufällige Bestimmung einer Schaltzeit im aktuellen Zustand, Auswahl der Transition mit der kürzesten Schaltzeit sowie Schalten dieser Transition mit Berechnung des Folgezustands) durchlaufen werden. Innerhalb eines Zustands werden die Werte der Leistungsparameter berechnet. Diese Werte werden zusammen mit der Verweildauer im Zustand an den Masterprozeß gesendet. Wichtige Komponenten des Simulationsprozesses sind die Berechnung von Schaltzeiten aus den Schaltzeitverteilungen mit einem Zufallszahlengenerator und die Erkennung der initialen transienten Simulationsphase. Diese Elemente sowie das Prinzip der verteilten Simulation mit Master- und Simulationsprozeß wurden aus dem Simulator von TimeNET übernommen.

Der Masterprozeß erhält die Werte der Leistungsparameter von den Simulationsprozessen und entscheidet, wann die Simulation beendet werden kann. Dazu schätzt er ständig Mittelwert und Varianz der Leistungsmaße, wofür die in TimeNET implementierten statistischen Verfahren verwendet werden. Das Konzept der Trennung von statistischer Auswertung und Simulation erleichtert die Anpassung des in [42] vorgestellten Simulators. Nach dem Erreichen der gewünschten Genauigkeit aller Leistungsparameter werden die Ergebnisse vom Masterprozeß in die Datei `net.RESULTS` geschrieben, von wo sie durch das Schnittstellenmodul der Oberfläche übernommen und dem Modellierer verfügbar gemacht werden.

7.3 Zusammenfassung

Für die praktische Anwendbarkeit einer Modellierungsmethode muß diese als Softwarewerkzeug realisiert sein. Dies ist für die in dieser Arbeit beschriebenen Techniken als Erweiterung des Modellierungs- und Bewertungswerkzeugs TimeNET geschehen, das für zeitbehaftete farblose Petri-Netze entwickelt wurde. Aufgrund der großen Unterschiede in den unterstützten Beschreibungsmitteln wurde eine neue Benutzungsoberfläche implementiert, die generisch an verschiedene Modellarten anpaßbar ist. Durch den modularen Aufbau der Analysekomponenten konnten einige der in TimeNET vorhandenen Programme für die Untersuchung der speziellen Petri-Netze eingesetzt werden.

Das vorgestellte Softwarewerkzeug ist unter den Betriebssystemen SunOS 4 (Solaris 1), SunOS 5 (Solaris 2) und DEC OSF1 lauffähig. Es setzt X11 Release 5 oder 6 voraus. Die neu implementierten Komponenten wurden in C++ realisiert, und die Benutzungsoberfläche verwendet das *Motif-Toolkit*. Informationen zur Installation, Konfigurierung und Aufruf der Benutzungsoberfläche können [43] entnommen werden.

Kapitel 8

Zusammenfassung und Ausblick

In dieser Dissertation wurde eine Modellierungs- und Analysemethode für Fertigungssysteme vorgestellt. Damit kann bereits während des Entwurfs das Verhalten, die Leistungsfähigkeit und Zuverlässigkeit einer Produktionsanlage untersucht werden. Mit dem Vergleich unterschiedlicher Varianten eines Fertigungsprozesses ist eine Optimierung möglich.

Die Voraussetzung dafür ist ein geeignetes Beschreibungsmittel. Petri-Netze sind grundsätzlich gut dazu geeignet, diskrete Fertigungssysteme mit ihren typischen Eigenschaften abzubilden. Die praktische Anwendbarkeit von farbigen Petri-Netzen als auch von solchen ohne unterscheidbare Marken ist jedoch in der Realität stark eingeschränkt. Für das betrachtete Anwendungsgebiet der Produktionstechnik wurde eine spezielle Art farbiger Petri-Netze definiert, die die Modellierung von Fertigungsprozessen erleichtert. Durch die Einschränkung der verwendbaren Markenfarben und eine entsprechende grafische Unterscheidung vereinfachen sich die Modelle. Es werden verschiedene Markenarten für die Modellierung von Werkstücken und die Darstellung von Betriebsmittelzuständen und anderen Informationen verwendet. Die Modellierung des typischen Verhaltens von Betriebsmitteln im Fertigungssystem wird durch eine spezielle Schaltsemantik unterstützt.

Mit diesen speziellen Petri-Netzen kann nun die Struktur des Fertigungssystems ähnlich dem realen Aufbau modelliert werden, was die Verständlichkeit erhöht. Das Strukturmodell beschreibt die Betriebsmittel und ihre arbeitsplanunabhängigen Eigenschaften. Es muß bei einem Wechsel des Fertigungsprogramms nicht geändert werden. Hierarchische Verfeinerung und die Verwendung von Teilmodellen aus einer Bibliothek von generischen Modellen erleichtern den Aufbau komplexer Modelle. Mit den speziellen Petri-Netzen werden neben der Struktur auch die Arbeitspläne modelliert. Die durchgängige und getrennte Modellierung beider Aspekte eines Fertigungssystems mit Petri-Netzen wird so ermöglicht. Eine Methode zur automatischen Erzeugung eines Gesamtmodells aus den Modellteilen wird vorgestellt. Dabei wird das Strukturmodell um sogenannte Übergangstabellen erweitert, in denen die möglichen Schaltvorgänge der Transitionen beschrieben werden.

Die verwendeten speziellen Petri-Netze können mit Methoden untersucht werden, die für allgemeine Petri-Netze entwickelt wurden. Es wird gezeigt, wie sich diese Techniken anwenden lassen, um qualitative und quantitative Eigenschaften eines Modells zu bestimmen. Durch die Ausnutzung der Einschränkungen in der Beschreibungsmethode vereinfachen sich die dazu notwendigen Berechnungen zum Teil. Qualitative Eigenschaften geben Auskunft über das

grundlegende Modellverhalten. Sie können oft anhand der Netzstruktur bestimmt werden und ermöglichen eine Überprüfung der Plausibilität eines Modells.

Um eine Leistungsbewertung durchzuführen, müssen zeitbehaftete Vorgänge im Modell abgebildet werden. Durch die dabei verwendbaren Schaltzeitverteilungen kann die numerische Analyse auf der Ebene des Erreichbarkeitsgraphen genauso wie für *extended deterministic and stochastic Petri nets* erfolgen. Falls die dabei zu beachtenden Einschränkungen nicht erfüllt sind oder der Erreichbarkeitsgraph zu groß für eine Analyse ist, wird diskrete Ereignissimulation eingesetzt. Mit beiden Techniken können die Werte von Leistungsparametern bestimmt und so die Leistungsfähigkeit und Zuverlässigkeit eines geplanten Fertigungssystems überprüft werden.

Viele Entwickler benutzen derzeit andere Beschreibungsmittel als Petri-Netze, um Fertigungssysteme zu modellieren. In der Arbeit wurde darum außerdem eine an Funktionssymbolen orientierte Beschreibungstechnik vorgestellt, deren Modelle sich automatisch in spezielle Petri-Netze umwandeln lassen. Über diesen Umweg sind die Bewertungstechniken dann auch für dieses Modellierungsmittel einsetzbar.

Die beschriebenen Methoden wurden in einem Softwarewerkzeug realisiert. Für die komfortable Eingabe von Modellen und ihre Untersuchung wurde eine anpassungsfähige Benutzungsoberfläche implementiert, die nicht auf die verwendeten speziellen Petri-Netze und Funktionssymbole eingeschränkt ist. Sie kann für viele aus Knoten und Kanten aufgebauten hierarchischen Modelle einfach parametrisiert und eingesetzt werden. Anhand der Modellierung und Untersuchung eines realen Anwendungsbeispiels wurde die Leistungsfähigkeit der entwickelten Methoden nachgewiesen.

Aus der Dissertation ergeben sich einige offene Probleme und Möglichkeiten weiterführender Arbeiten. Die während des Entwurfs eines Fertigungssystems auftretenden Schwierigkeiten gelten in ähnlicher Weise für die Entwicklung anderer komplexer technischer Einrichtungen wie Kommunikations- oder Verkehrsleitsysteme. Es liegt nahe, für diese Anwendungsgebiete ebenfalls speziell angepaßte Petri-Netze und entsprechende Untersuchungsmethoden zu entwickeln. Möglicherweise lassen sich dadurch ähnliche Verbesserungen bei der praktischen Anwendbarkeit von auf Petri-Netzen basierenden Modellierungs- und Analysetechniken erzielen.

Fertigungssysteme enthalten typischerweise eine Mischung aus deterministischen (Transport- und Bearbeitungszeiten) sowie annäherungsweise exponentialverteilten Schaltzeiten (Ausfallraten). Die derzeit verfügbaren numerischen Analysetechniken sind strukturell auf eine schaltfähige Transition mit nicht exponentieller Schaltzeit beschränkt. Dies ist für praktische Anwendungen oft nicht ausreichend, in denen der Einfluß der Schaltzeitverteilungen nicht vernachlässigbar ist. Eine Erweiterung der Modellierungsmächtigkeit in dieser Richtung und die Entwicklung entsprechender numerischer Analysetechniken bleibt ein wichtiges Forschungsthema. Bei der Verwendung von stochastischen Petri-Netzen mit diskreter Zeitachse bestehen die genannten Einschränkungen nicht. Leider wird dies mit einer weiteren Vergrößerung des Zustandsraums erkauft. Die Größe der zu untersuchenden Erreichbarkeitsgraphen ist ein zentrales Problem der numerischen Analyse von Petri-Netzen. Verschiedene Arbeiten beschäftigen sich mit Methoden zur strukturierten Analyse, wobei der Zustandsraum nicht mehr vollständig aufgestellt werden muß. Ein anderer Ansatz verfolgt die iterative näherungsweise Berechnung der Aufenthaltswahrscheinlichkeiten eines Petri-Netzes. Es ist

zu prüfen, inwiefern diese Arbeiten auf die vorgestellte Netzklasse anwendbar sind bzw. wie sich ihre speziellen Eigenschaften dabei ausnutzen lassen.

Neben den methodischen Arbeiten wird das vorgestellte prototypische Softwarewerkzeug weiterentwickelt. Dabei ist eine effiziente Implementierung der Algorithmen wichtig. Ein wesentlicher Gesichtspunkt für die Anwendbarkeit eines solchen Werkzeugs sind die Antwortzeiten bis zur Berechnung der interessierenden Fragestellungen. Um ein Modell intuitiv verständlich zu machen, können die grafischen Fähigkeiten der Benutzungsoberfläche weiter verbessert werden. Dazu wäre die Visualisierung des Verhaltens mit verkleinerten Darstellungen realer Maschinen o.ä. möglich, wie sie in Simulationswerkzeugen verwendet wird.

Verschiedene wichtige Problemfelder der Produktionsplanung und -steuerung waren nicht Gegenstand dieser Arbeit. Dazu gehören unter anderem die Feinplanung, Kostenrechnung und Fertigungssteuerung. Die verwendeten Petri-Netze sind möglicherweise nach entsprechenden Erweiterungen auch für die Darstellung und Untersuchung dieser Problemstellungen einsetzbar. Für die Feinplanung könnte ein externer Baustein eingesetzt werden, der anhand des aktuellen Zustands des Modells Entscheidungen trifft. Diese wirken sich dann auf die Schaltfähigkeit von Transitionen aus. Auf diese Weise wäre eine Untersuchung von *online*-Feinplanungsstrategien und ihrer Leistungsfähigkeit möglich. Um die Kosten eines Fertigungsprozesses zu bestimmen, ist vor allem eine Erweiterung der bisher verfügbaren Konstrukte für Leistungsparameter notwendig.

Ein Problem bei der Realisierung einer Fertigungssteuerung anhand eines Modells stellt seine Kopplung mit dem realen Fertigungsprozeß dar. Diese muß ohne größere Änderungen am Formalismus der Petri-Netze möglich sein und setzt für nichttriviale Systeme eine Integration der Fertigungssteuerung in das Modell voraus. Mit dem Einsatz des Petri-Netz-Modells eines Fertigungssystems während des Entwurfs, der Untersuchung, Feinplanung und Steuerung kann so eine durchgängige Unterstützung mit demselben Modell realisiert werden.

A Symbole und Abkürzungen

Es folgt ein Verzeichnis der in der Arbeit verwendeten Abkürzungen und Symbole. Wenn die Bedeutung in der Mehrzahl steht, handelt es sich um eine Menge.

A	Kanten
AZ	Anschlußzuweisungsfunktion
A^E	Elementarkanten
A^Φ	Produktkanten
a	eine Kante
B	Bearbeitungsstufen
\mathbf{B}	$\{\text{Wahr, Falsch}\}$
\mathcal{B}_m	Grundmenge, über dem die Multimenge m definiert ist
C	Farbfunktion
CPN	<i>colored petri net</i>
c	eine Farbe bzw. Markentyp
DSPN	<i>deterministic and stochastic Petri net</i>
E	Elementarfarbmenge $\{e\}$
ES	Schaltsemantik <i>exclusive server</i>
eDSPN	<i>extended deterministic and stochastic Petri net</i>
\mathcal{F}	Menge der Schaltzeitverteilungen
G	Schaltbedingungsfunktion (<i>guard</i>)
GSPN	<i>generalized stochastic Petri net</i>
I	Initialisierungsfunktion
IS	Schaltsemantik <i>infinite server</i>
J	Inzidenzmatrix
K	Kapazitätsfunktion
MME	<i>marking mutual exclusive</i>
\mathcal{M}_A	Menge der Multimengen über A
m/S	Beschränkung der Multimenge m auf Elemente aus S
\mathbb{N}	Menge der natürlichen Zahlen
P	Stellen
P^A	Anschlußstellen
\mathcal{P}^A	Menge der Teilmengen von A (Potenzmenge)
P^E	Elementarstellen
P^N	nicht-Anschlußstellen
P^{in}	Eingangsstellen (einer Transition)
P^{out}	Ausgangsstellen (einer Transition)
P^Φ	Produktstellen
PE	Stellenelemente
PV	Vorauswahl-Funktion
PV_0	initiale Vorauswahl-Funktion
p	eine Stelle
\mathbb{R}	Menge der reellen Zahlen
\mathbb{R}^+	Menge der positiven reellen Zahlen
R	Markierungen

RG	Erreichbarkeitsgraph
RS	Erreichbarkeitsmenge
r	eine Markierung
r_0	Initialmarkierung
S	Seiten
SME	<i>structural mutual exclusive</i>
SPN	<i>stochastic Petri net</i>
SS	Schaltsemantik
SZ	Seitenzuweisungsfunktion
s	eine Seite
s_p	Primärseite
sCPN	spezielles Petri-Netz
T	Transitionen
\mathcal{T}_m	Träger der Multimenge m
T^N	nicht-Vertretertransitionen
T^V	Vertretertransitionen
T^{imm}	zeitlose Transitionen
T^{tim}	zeitbehaftete Transitionen
t	eine Transition
U	Übergangsfunktion
UE	Übergangselemente
U^{exp}	exponentielle Übergänge
U^{gen}	nichtexponentielle Übergänge
u	ein Übergang
u^{delay}	Schaltzeitverteilung des Übergangs u
u^{guard}	Schaltbedingung des Übergangs u
u^{in}	Eingangsverhalten des Übergangs u
u^{out}	Ausgangsverhalten des Übergangs u
u^{select}	Auswahlwahrscheinlichkeit des Übergangs u
u^{weight}	Gewicht des Übergangs u
V	Versionen
$v_{i,e}$	Wert des Ausdrucks e in Markierung r_i
W	Produkte
Z	zulässige Zustände
Z^{abs}	absorbierende Zustände
Z^{exp}	exponentielle Zustände
Z^{imm}	verschwindende Zustände
Z^{int}	Zwischenzustände
Z^{tim}	andauernde Zustände
z	ein Zustand
z_0	initialer Zustand
Θ	Schrittmenge
Θ^*	schaltfähige Schrittmenge
θ	ein Schritt
Σ	Menge der Farbmengen

Φ	Produktfahrbmenge
ψ	Invariante
ψ^*	Invariante mit minimalem Träger
Ψ	Invarianten
Ψ^*	Invarianten mit minimalem Träger
$ m $	Mächtigkeit der Multimenge m
\overline{m}	Entfaltung von m

B Multimengen

Multimengen sind Mengen, in denen die Elemente mehrfach auftreten können. Sie sind formal wie folgt definiert: Eine **Multimenge** (*multi set, bag*) m über einer endlichen, nichtleeren Menge \mathcal{B}_m ist eine Funktion $m : \mathcal{B}_m \rightarrow \mathbb{N}$, die jedem Element ihrer **Grundmenge** \mathcal{B}_m eine natürliche Zahl zuordnet. Der Funktionswert $m(x) \in \mathbb{N}$ beschreibt die Anzahl bzw. den Koeffizient des Elementes $x \in \mathcal{B}_m$ in m . Ein Multiset m läßt sich als formale Summe darstellen: $m = \sum_{b \in \mathcal{B}_m} m(b)'b$. Werden Multimengen in Summenform aufgeschrieben, können die Summanden mit $m(b) = 0$ weggelassen werden, wenn die Grundmenge \mathcal{B}_m eindeutig bestimmt ist. Außerdem dürfen für Elemente einer Multimenge, deren Koeffizient gleich 1 ist, der Koeffizient und das '-Zeichen weggelassen werden. Der **Träger** (*support*) \mathcal{T}_m eines Multisets m ist die Menge der Elemente aus \mathcal{B}_m , deren Koeffizient größer als Null ist: $\mathcal{T}_m = \{t \in \mathcal{B}_m : m(t) > 0\}$. Für leere Multimengen kann ebenso wie für leere Mengen das Symbol \emptyset geschrieben werden. Die Menge aller Multimengen über einer Grundmenge S wird mit \mathcal{M}_S bezeichnet. Folgende Operationen und Relationen sind für Multimengen definiert:

- Die **Summe** zweier Multimengen m_1 und m_2 , $(m_1 + m_2) : \mathcal{B}_{m_1} \cup \mathcal{B}_{m_2} \rightarrow \mathbb{N}$, ist wie folgt definiert:

$$\forall b \in \mathcal{B}_{m_1} \cup \mathcal{B}_{m_2} : (m_1 + m_2)(b) = \begin{cases} m_1(b) & \text{falls } b \in \mathcal{B}_{m_1} \setminus \mathcal{B}_{m_2} \\ m_2(b) & \text{falls } b \in \mathcal{B}_{m_2} \setminus \mathcal{B}_{m_1} \\ m_1(b) + m_2(b) & \text{sonst} \end{cases}$$

- Die **Subtraktion** zweier Multimengen m_1 und m_2 , $(m_1 - m_2) : \mathcal{B}_{m_1} \rightarrow \mathbb{N}$, ist wie folgt definiert:

$$\forall b \in \mathcal{B}_{m_1} : (m_1 - m_2)(b) = \begin{cases} m_1(b) & \text{falls } b \in \mathcal{B}_{m_1} \setminus \mathcal{B}_{m_2} \\ \max(0, m_1(b) - m_2(b)) & \text{sonst} \end{cases}$$

- Eine Multimenge m kann mit einem **Skalar** $n \in \mathbb{N}$ **multipliziert** werden ($n * m$) : $\mathcal{B}_m \rightarrow \mathbb{N}$:

$$\forall b \in \mathcal{B}_m : (n * m)(b) = n * m(b)$$

- **Vergleiche** zwischen Multimengen werden auf ihre Koeffizienten zurückgeführt:

$$m_1 > m_2 \Leftrightarrow \forall b \in \mathcal{B}_{m_2} : \begin{cases} m_2(b) = 0 & \text{falls } b \in \mathcal{B}_{m_2} \setminus \mathcal{B}_{m_1} \\ m_1(b) > m_2(b) & \text{falls } b \in \mathcal{B}_{m_1} \cap \mathcal{B}_{m_2} \end{cases}$$

$$m_1 = m_2 \Leftrightarrow \forall b \in \mathcal{B}_{m_1} \cup \mathcal{B}_{m_2} : \begin{cases} m_1(b) = 0 & \text{falls } b \in \mathcal{B}_{m_1} \setminus \mathcal{B}_{m_2} \\ m_2(b) = 0 & \text{falls } b \in \mathcal{B}_{m_2} \setminus \mathcal{B}_{m_1} \\ m_1(b) = m_2(b) & \text{sonst} \end{cases}$$

$$m_1 \leq m_2 \Leftrightarrow m_1 = m_2 \vee m_1 < m_2$$

$$m_1 \geq m_2 \Leftrightarrow m_1 = m_2 \vee m_1 > m_2$$

- Ein Element b aus der Grundmenge \mathcal{B}_m einer Multimenge m kann als **Element der Multimenge** bezeichnet werden:

$$\forall b \in \mathcal{B}_m : b \in m \Leftrightarrow m(b) > 0$$

- Als **Mächtigkeit** einer Multimenge m wird die Summe ihrer Koeffizienten bezeichnet:

$$|m| = \sum_{b \in \mathcal{B}_m} m(b)$$

- Eine Multimenge m kann mit einer Menge $S \neq \emptyset$ auf m/S , $(m/S) : S \rightarrow \mathbf{N}$ **beschränkt** werden:

$$\forall b \in S : (m/S)(b) = \begin{cases} m(b) & \text{falls } b \in S \\ 0 & \text{sonst} \end{cases}$$

C Übergangstabellen und Ausdrücke

In den folgenden Abschnitten des Anhangs wird die Syntax verschiedener Beschreibungselemente in Backus-Naur-Form definiert. Nichtterminale Symbole stehen in eckigen Klammern $\langle \text{symbol} \rangle$, während terminale fett gedruckt sind: **symbol**. Es werden die üblichen Zeichen für Alternative, optionales und mehrfaches Auftreten verwendet.

Die folgende Syntaxdefinition beschreibt den Aufbau einer Übergangstabelle.

$$\begin{aligned} \langle \text{transition table} \rangle &\rightarrow \mathbf{[\langle \text{table entry} \rangle \{ | \langle \text{table entry} \rangle \}]} \\ \langle \text{table entry} \rangle &\rightarrow \mathbf{[\langle \text{local condition} \rangle] \langle \text{arc bindings} \rangle ; [\langle \text{select} \rangle] ; [\langle \text{weight} \rangle] ; \langle \text{delay} \rangle} \\ \langle \text{arc bindings} \rangle &\rightarrow \langle \text{arc binding} \rangle \{ , \langle \text{arc binding} \rangle \} \\ \langle \text{arc binding} \rangle &\rightarrow \mathbf{(\text{in}\#\langle \text{place name} \rangle | \text{out}\#\langle \text{place name} \rangle) = \langle \text{mset expr} \rangle} \\ \langle \text{select} \rangle &\rightarrow \langle \text{real expr} \rangle \\ \langle \text{weight} \rangle &\rightarrow \langle \text{real expr} \rangle \\ \langle \text{local condition} \rangle &\rightarrow \mathbf{(\langle \text{logic condition} \rangle)} \end{aligned}$$

Innerhalb eines Übergangs wird die Dichtefunktion der Schaltzeitverteilung $\langle \text{delay} \rangle$ spezifiziert. Neben den einfachen Formen für exponentielle, deterministische und gleichverteilte können auch expolynomiale Funktionen nach folgender Syntax angegeben werden [32]. Bei allen impliziten Multiplikationen innerhalb einer Funktionsdefinition kann auch ein Multiplikationszeichen $*$ verwendet werden. Zur Vereinfachung werden hier keine Einschränkungen des Wertebereichs von Ausdrücken und Konstanten angegeben.

$$\begin{aligned}
\langle \text{delay} \rangle &\rightarrow \mathbf{IMMEDIATE} \\
&| \mathbf{EXPONENTIAL}(\langle \text{real expr} \rangle) \\
&| \mathbf{DETERMINISTIC}(\langle \text{real expr} \rangle) \\
&| \mathbf{UNIFORM}(\langle \text{real const} \rangle, \langle \text{real const} \rangle) \\
&| \langle \text{pmf expr} \rangle \\
\langle \text{pmf expr} \rangle &\rightarrow \langle \text{pmf expr} \rangle + \langle \text{pmf expr} \rangle \\
&| \langle \text{pmf expr} \rangle - \langle \text{pmf expr} \rangle \\
&| \langle \text{impulse} \rangle \\
&| \langle \text{exp rect} \rangle \\
\langle \text{impulse} \rangle &\rightarrow [\langle \text{real const} \rangle] \mathbf{I} [\langle \text{real const} \rangle] \\
\langle \text{exp rect} \rangle &\rightarrow [(\langle \text{expoly expr} \rangle)] \mathbf{R} [\langle \text{real const} \rangle, \langle \text{real const} \rangle] \\
\langle \text{expoly expr} \rangle &\rightarrow \langle \text{expoly expr} \rangle + \langle \text{expoly expr} \rangle \\
&| \langle \text{expoly expr} \rangle - \langle \text{expoly expr} \rangle \\
&| (\langle \text{expoly expr} \rangle) \\
&| \langle \text{expoly} \rangle \\
\langle \text{expoly} \rangle &\rightarrow [\langle \text{real const} \rangle] [\mathbf{x} [\wedge \langle \text{int const} \rangle]] [\mathbf{e}^{\langle - \langle \text{real const} \rangle \mathbf{x} \rangle}]
\end{aligned}$$

In den Übergangstabellen, den Parameterdefinitionen usw. treten innerhalb der Netzdefinition Ausdrücke verschiedener Art auf. Ihre Syntax wird mit der folgenden Definition angegeben. Dabei können reellwertige, ganzzahlige und boolesche Ausdrücke bzw. Konstanten sowie solche vom Typ Multimenge verwendet werden. Zur kompakten Darstellung sind Einschränkungen bei der Verwendung bestimmter Konstrukte angegeben. Diese betreffen Definitionen von Leistungsparametern sowie markierungsabhängige Ausdrücke.

$$\begin{aligned}
\langle \text{real expr} \rangle &\rightarrow \langle \text{real const} \rangle \\
&| \langle \text{int const} \rangle \\
&| \langle \text{token count} \rangle \text{ (falls markierungsabhängig erlaubt)} \\
&| - \langle \text{real expr} \rangle \\
&| (\langle \text{real expr} \rangle) \\
&| \langle \text{real expr} \rangle (+ | - | * | /) \langle \text{real expr} \rangle \\
&| \langle \text{reward expr} \rangle \text{ (nur für Leistungsparameter, nicht geschachtelt)} \\
\langle \text{reward expr} \rangle &\rightarrow \mathbf{E} [\langle \text{real expr} \rangle] \\
&| \mathbf{E} [\langle \text{real expr} \rangle | \langle \text{logic condition} \rangle] \\
&| \mathbf{P} [\langle \text{logic condition} \rangle] \\
&| \mathbf{P} [\langle \text{logic condition} \rangle | \langle \text{logic condition} \rangle] \\
&| \mathbf{TP} [\langle \text{transition name} \rangle] \\
\langle \text{logic condition} \rangle &\rightarrow \langle \text{comparison} \rangle \\
&| ! \langle \text{logic condition} \rangle \\
&| (\langle \text{logic condition} \rangle) \\
&| \langle \text{logic condition} \rangle | \langle \text{logic condition} \rangle \\
&| \langle \text{logic condition} \rangle \& \langle \text{logic condition} \rangle \\
&| \mathbf{FALSE} | \mathbf{TRUE} \\
\langle \text{comparison} \rangle &\rightarrow \langle \text{real expr} \rangle \langle \text{comp operator} \rangle \langle \text{real expr} \rangle \\
&| \langle \text{mset expr} \rangle \langle \text{comp operator} \rangle \langle \text{mset expr} \rangle \\
\langle \text{comp operator} \rangle &\rightarrow = | \neq | > | < | \geq | \leq
\end{aligned}$$

$\langle \text{mset expr} \rangle$	\rightarrow	$\langle \text{p mset} \rangle$ $ \langle \text{e mset} \rangle$ $ \langle \text{place contents} \rangle$ (<i>falls markierungsabhängig erlaubt</i>) $ (\langle \text{mset expr} \rangle)$ $ \langle \text{mset expr} \rangle (+ -) \langle \text{mset expr} \rangle$ $ \langle \text{real expr} \rangle * \langle \text{mset expr} \rangle$ $ \langle \text{mset expr} \rangle / \langle \text{real expr} \rangle$
$\langle \text{p mset} \rangle$	\rightarrow	$[(\langle \text{real expr} \rangle) *] \langle \text{p mset const} \rangle$ $ \{ \langle \text{p mset const} \rangle \{ , \langle \text{p mset const} \rangle \} \}$
$\langle \text{e mset} \rangle$	\rightarrow	$[(\langle \text{real expr} \rangle) *] \langle \text{e mset const} \rangle$ $ \{ \langle \text{e mset const} \rangle \{ , \langle \text{e mset const} \rangle \} \}$
$\langle \text{token count} \rangle$	\rightarrow	# $\langle \text{place name} \rangle [(\langle \text{mset expr} \rangle)]$
$\langle \text{place contents} \rangle$	\rightarrow	@ $\langle \text{place name} \rangle [(\langle \text{mset expr} \rangle)]$
$\langle \text{p mset const} \rangle$	\rightarrow	$[\langle \text{int const} \rangle '] \langle \text{part} \rangle$
$\langle \text{e mset const} \rangle$	\rightarrow	$[\langle \text{int const} \rangle '] \mathbf{e}$
$\langle \text{part} \rangle$	\rightarrow	$\langle \text{part name} \rangle \langle \text{part version} \rangle \langle \text{part step} \rangle$ $ \langle \text{part name} \rangle ?$
$\langle \text{part name} \rangle$	\rightarrow	$\langle \text{identifier} \rangle$
$\langle \text{part version} \rangle$	\rightarrow	$[(\langle \text{int const} \rangle \langle \text{identifier} \rangle ?)] .$
$\langle \text{part step} \rangle$	\rightarrow	$\langle \text{int const} \rangle \langle \text{identifier} \rangle$
$\langle \text{place name} \rangle$	\rightarrow	$\langle \text{net object} \rangle$
$\langle \text{transition name} \rangle$	\rightarrow	$\langle \text{net object} \rangle$
$\langle \text{page name} \rangle$	\rightarrow	$\langle \text{net object} \rangle$
$\langle \text{net object} \rangle$	\rightarrow	$[\backslash] \langle \text{identifier} \rangle \{ \backslash \langle \text{identifier} \rangle \}$
$\langle \text{identifier} \rangle$	\rightarrow	$\langle \text{letter} \rangle \{ \langle \text{letter} \rangle \langle \text{digit} \rangle - \}$
$\langle \text{letter} \rangle$	\rightarrow	$\mathbf{a} .. \mathbf{z} \mathbf{A} .. \mathbf{Z}$
$\langle \text{real const} \rangle$	\rightarrow	$\langle \text{digit} \rangle \{ \langle \text{digit} \rangle \} . \{ \langle \text{digit} \rangle \} [\langle \text{exponent} \rangle]$ $ \{ \langle \text{digit} \rangle \} . \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \} [\langle \text{exponent} \rangle]$ $ \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \} \langle \text{exponent} \rangle$
$\langle \text{exponent} \rangle$	\rightarrow	$(\mathbf{E} \mathbf{e}) [(+ -)] \langle \text{digit} \rangle \{ \langle \text{digit} \rangle \}$
$\langle \text{int const} \rangle$	\rightarrow	$\langle \text{digit} \rangle \{ \langle \text{digit} \rangle \}$
$\langle \text{digit} \rangle$	\rightarrow	$\mathbf{0} .. \mathbf{9}$

D Formate von Modellbeschreibungen

Im Anschluß wird die Syntax einer Netzklassenbeschreibung angegeben, die in einer Netzklassendatei steht [43]. Einige der nicht weiter erklärten nichtterminalen Symbole sind nach ihrer Bedeutung benannt. Die Symbole $\langle \text{int-const} \rangle$ und $\langle \text{fp-const} \rangle$ stehen hier für ganzzahlige und reellwertige Zahlen.

⟨netcl-file⟩	→	(netclass subclass) [{ ⟨netcl-def⟩ ; }]
⟨netcl-def⟩	→	⟨name-def⟩ ⟨attr-def⟩ (⟨nodecl-list⟩ ; ⟨conncl-list⟩)
⟨nodecl-list⟩	→	[{ ⟨nodecl-def⟩ ; }]
⟨conncl-list⟩	→	[{ ⟨conncl-def⟩ ; }]
⟨nodecl-def⟩	→	⟨objcl-def⟩ ⟨subcl-def⟩
⟨conncl-def⟩	→	⟨objcl-def⟩ ⟨rule-list⟩
⟨objcl-def⟩	→	⟨name-def⟩ ⟨group-def⟩ ⟨prefix-def⟩ ⟨rels-list⟩ ⟨attr-list⟩ ⟨shape-list⟩
⟨rule-list⟩	→	rules = [{ ⟨rule-def⟩ ; }]
⟨rule-def⟩	→	[⟨anchor-name⟩ , ⟨anchor-name⟩]
⟨anchor-name⟩	→	⟨nodeclass-name⟩ ⟨nodegroup-name⟩
⟨subcl-def⟩	→	subclass = ⟨subclass-name⟩
⟨name-def⟩	→	name = ⟨ident⟩
⟨group-def⟩	→	group = ⟨ident⟩
⟨prefix-def⟩	→	nameprefix = ⟨ident⟩
⟨rels-list⟩	→	relatives = [{ ⟨position⟩ ; }]
⟨attr-list⟩	→	attributes = [{ ⟨attr-def⟩ ; }]
⟨shape-list⟩	→	shape = [{ ⟨shape-def⟩ ; }]
⟨shape-def⟩	→	⟨shape-type⟩ [⟨context-def⟩ ⟨shape-params⟩]
⟨attr-def⟩	→	⟨data-type⟩ ⟨ident⟩ [= ⟨initial-value⟩]
⟨data-type⟩	→	⟨bool-type⟩ ⟨int-type⟩ ⟨float-type⟩ ⟨enum-type⟩ ⟨text-type⟩ ⟨ident-type⟩ ⟨struct-type⟩ ⟨list-type⟩
⟨shape-type⟩	→	⟨pline-type⟩ ⟨ellarc-type⟩ ⟨label-type⟩ ⟨compo-type⟩
⟨bool-type⟩	→	bool [[⟨comment⟩ [, ⟨comment⟩]]]
⟨int-type⟩	→	int [[⟨int-const⟩ [, ⟨int-const⟩]]]
⟨float-type⟩	→	float [[⟨fp-const⟩ [, ⟨fp-const⟩]]]
⟨enum-type⟩	→	enum [⟨ident⟩ { , ⟨ident⟩ }]
⟨text-type⟩	→	text [[⟨int-const⟩]]
⟨ident-type⟩	→	ident [[(true false)]]
⟨struct-type⟩	→	struct [{ ⟨attr-def⟩ ; }]
⟨list-def⟩	→	list [⟨data-type⟩]
⟨pline-type⟩	→	pline [⟨context⟩ { , ⟨position⟩ }]
⟨label-type⟩	→	label [⟨context⟩ , ⟨position⟩ , ⟨context-expr⟩ [, ⟨alignment⟩]]
⟨ellarc-type⟩	→	ellarc [⟨context⟩ , ⟨position⟩ , ⟨x-radius⟩ [, ⟨y-radius⟩ [, ⟨beg-angle⟩ , ⟨end-angle⟩]]]
⟨compo-type⟩	→	compo [⟨context⟩ , ⟨position⟩ { , ⟨shape-def⟩ }]
⟨context⟩	→	[{ ⟨context-def⟩ ; }]
⟨context-def⟩	→	(⟨context-var⟩ ⟨ident⟩) = ⟨context-expr⟩
⟨context-var⟩	→	thick color font spline filled dashed cond cpes
⟨alignment⟩	→	left center right
⟨ident⟩	→	[⟨comment⟩] a-zA-Z_ { a-zA-Z0-9_ }
⟨comment⟩	→	"..."
⟨position⟩	→	[⟨fp-const⟩ , ⟨fp-const⟩]

Ein Modell einer Netzklasse wird in einer Netzmodell-Datei abgespeichert. Deren Format [43] wird im folgenden beschrieben. Über die Symbole $\langle \text{netclass-name} \rangle$, $\langle \text{nodeclass-name} \rangle$ sowie $\langle \text{connclass-name} \rangle$ wird der Bezug zu den Definitionen aus der Netzklassendatei hergestellt. Das Symbol $\langle \text{id-const} \rangle$ steht für eine Identifikator-Konstante, die ein Netzobjekt referenziert.

$\langle \text{net-file} \rangle \rightarrow \langle \text{netclass-name} \rangle (\{ \langle \text{net-def} \rangle ; \})$
 $\langle \text{net-def} \rangle \rightarrow \langle \text{nattr-asn} \rangle | \langle \text{obj-list} \rangle$
 $\langle \text{obj-list} \rangle \rightarrow \mathbf{objects} = (\{ \langle \text{node-def} \rangle ; \} \{ \langle \text{conn-def} \rangle ; \})$
 $\langle \text{node-def} \rangle \rightarrow \langle \text{nodeclass-name} \rangle (\{ \langle \text{node-asn} \rangle ; \})$
 $\langle \text{conn-def} \rangle \rightarrow \langle \text{connclass-name} \rangle (\{ \langle \text{conn-asn} \rangle ; \})$
 $\langle \text{node-asn} \rangle \rightarrow \langle \text{obj-def} \rangle | \langle \text{real-asn} \rangle | \langle \text{angle-asn} \rangle | \langle \text{pos-asn} \rangle$
 $\langle \text{conn-asn} \rangle \rightarrow \langle \text{obj-def} \rangle | \langle \text{inp-asn} \rangle | \langle \text{outp-asn} \rangle | \langle \text{npos-asn} \rangle$
 $\langle \text{obj-def} \rangle \rightarrow \langle \text{name-asn} \rangle | \langle \text{id-asn} \rangle | \langle \text{rels-asn} \rangle | \langle \text{nattr-asn} \rangle | \langle \text{scale-asn} \rangle$
 $\langle \text{name-asn} \rangle \rightarrow \mathbf{name} = \langle \text{id-const} \rangle$
 $\langle \text{rels-asn} \rangle \rightarrow \mathbf{relatives} = (\{ \langle \text{position} \rangle ; \})$
 $\langle \text{nattr-asn} \rangle \rightarrow \mathbf{attributes} = (\{ \langle \text{attr-asn} \rangle ; \})$
 $\langle \text{attr-asn} \rangle \rightarrow \langle \text{id-const} \rangle = \langle \text{attribute-value} \rangle$
 $\langle \text{pos-asn} \rangle \rightarrow \mathbf{position} = \langle \text{position} \rangle$
 $\langle \text{scale-asn} \rangle \rightarrow \mathbf{scale} = \langle \text{position} \rangle$
 $\langle \text{angle-asn} \rangle \rightarrow \mathbf{orientation} = \langle \text{fp-const} \rangle$
 $\langle \text{npos-asn} \rangle \rightarrow \mathbf{positions} = (\{ \langle \text{position} \rangle ; \})$
 $\langle \text{id-asn} \rangle \rightarrow \mathbf{id} = \langle \text{id-const} \rangle$
 $\langle \text{real-asn} \rangle \rightarrow \mathbf{real} = \langle \text{id-const} \rangle$
 $\langle \text{inp-asn} \rangle \rightarrow \mathbf{input} = \langle \text{id-const} \rangle$
 $\langle \text{outp-asn} \rangle \rightarrow \mathbf{output} = \langle \text{id-const} \rangle$
 $\langle \text{position} \rangle \rightarrow (\langle \text{fp-const} \rangle , \langle \text{fp-const} \rangle)$

Um die grafische Darstellung abhängig von den Attributen eines Objektes zu machen, werden sogenannte Kontext-Ausdrücke benötigt [43].

$\langle \text{context-expr} \rangle \rightarrow \langle \text{q-op} \rangle$
 $\langle \text{q-op} \rangle \rightarrow \langle \text{or} \rangle [? \langle \text{q-op} \rangle : \langle \text{or} \rangle]$
 $\langle \text{or} \rangle \rightarrow \langle \text{and} \rangle \{ | \langle \text{and} \rangle \}$
 $\langle \text{and} \rangle \rightarrow \langle \text{comp} \rangle \{ \& \langle \text{comp} \rangle \}$
 $\langle \text{comp} \rangle \rightarrow \langle \text{expr} \rangle [> | < | = \langle \text{expr} \rangle]$
 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle | \{ + | - \langle \text{term} \rangle \}$
 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle | \{ * | / \langle \text{factor} \rangle \}$
 $\langle \text{factor} \rangle \rightarrow [! | -] \langle \text{ref} \rangle | \langle \text{const} \rangle | (\langle \text{q-op} \rangle)$
 $\langle \text{ref} \rangle \rightarrow (@ \langle \text{id-const} \rangle [\langle \text{q-op} \rangle] | \{ . \langle \text{id-const} \rangle \}) | \# \langle \text{sys-ref} \rangle$
 $\langle \text{sys-ref} \rangle \rightarrow \mathbf{name} | \mathbf{selected} | \mathbf{virtual} | \mathbf{draft} | \mathbf{mono}$
 $\langle \text{const} \rangle \rightarrow \langle \text{text} \rangle | \langle \text{id-const} \rangle | \langle \text{fp-const} \rangle$
 $\langle \text{text} \rangle \rightarrow \dots$
 $\langle \text{id-const} \rangle \rightarrow \mathbf{a-zA-Z_} \{ \mathbf{a-zA-Z0-9_} \}$

Literaturverzeichnis

- [1] M. Ajmone Marsan, G. Balbo, G. Chiola, G. Conte, S. Donatelli, und G. Francheschinis, An Introduction to Generalized Stochastic Petri Nets, *Microelectronics and Reliability, Special Issue on Petri Nets* (1989) 1–36.
- [2] M. Ajmone Marsan, Stochastic Petri Nets: An Elementary Introduction, in: G. Rozenberg, Ed., *Advances in Petri Nets 1989, Lecture Notes in Computer Science*, Bd. 424 (Springer Verlag, 1990) 1–29.
- [3] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, und A. Cumani, The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets, *IEEE Transactions on Software Engineering* **15** (7) (1989) 832–846.
- [4] M. Ajmone Marsan, G. Balbo, und G. Conte, A class of Generalized Stochastic Petri Nets for the performance evaluation of multiprocessor systems, *ACM Transactions on Computer Systems* **2** (2) (1984) 93–122.
- [5] M. Ajmone Marsan und G. Chiola, On Petri Nets with Deterministic and Exponentially Distributed Firing Times, in: G. Rozenberg, Ed., *Advances in Petri Nets 1987, Lecture Notes in Computer Science*, Bd. 266 (Springer Verlag, 1987) 132–145.
- [6] R. Y. Al-Jaar und A. A. Desrochers, Performance Evaluation of Automated Manufacturing Systems Using Generalized Stochastic Petri Nets, *IEEE Transactions on Robotics and Automation* **6** (6) (1990) 621–639.
- [7] H. Alla, P. Ladet, J. Martínez, und M. Silva, Modeling and validation of complex systems by coloured Petri nets, Application to a Flexible Manufacturing System, in: G. Rozenberg, Ed., *Advances in Petri Nets 1984, Lecture Notes in Computer Science*, Bd. 188 (Springer Verlag, 1985) 15–31.
- [8] G. Balbo, G. Chiola, G. Franceschinis, und G. Molinar Roet, On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic Petri Net Models, in: *Proc. 2nd Int. Workshop on Petri Nets and Performance Models* (Madison, Wisconsin, 1987) 136–145.
- [9] G. Balbo, G. Chiola, G. Franceschinis, und G. Molinar Roet, Generalized Stochastic Petri Nets for the Performance Evaluation of FMS, in: *Proc. Int. Conf. on Robotics and Automation* (Raleigh, North Carolina, 1987) 1013–1018.

- [10] C. L. Beck und B. H. Krogh, Models for Simulation and Discrete Control of Manufacturing Systems, in: *Proc. Int. Conf. on Robotics and Automation* (1986) 305–310.
- [11] W. Berndt, Anforderungen an die Simulation des dynamischen Verhaltens von Produktionsanlagen aus der Sicht der Automobilindustrie und Lösungswege zur Simulation, in: *Proc. Techno Congress, 5. Fachtagung* (Sindelfingen, 1992) .
- [12] A. Blakemore, The Cost of Eliminating Vanishing Markings from Generalized Stochastic Petri Nets, in: *Proc. 3rd Int. Workshop on Petri Nets and Performance Models* (Kyoto, Japan, 1989) 85–92.
- [13] S. Bode, Entwicklung einer Methode zur Generierung von Petri-Netz-Modellen für Fertigungssysteme, Diplomarbeit, Technische Universität Berlin, Februar 1995.
- [14] D. Y. Chao und D. T. Wang, XPN-FMS: A CAD Tool for FMS Modeling, Analysis, Animation, and Simulation Using Petri Nets and X Window, *Int. J. of Flexible Manufacturing* (7) (1995) 339–360.
- [15] G. Chiola, GreatSPN 1.5 Software Architecture, in: *Proc. 5th Int. Conf. on Modeling Techniques and Tools for Performance Evaluation* (Torino, Italy, 1991) 117–132.
- [16] G. Chiola, M. Ajmone Marsan, G. Balbo, und G. Conte, Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications, *IEEE Transactions on Software Engineering* **19** (2) (1993) 89–107.
- [17] H. Choi, V. G. Kulkarni, und K. S. Trivedi, Markov Regenerative Stochastic Petri Nets, *Performance Evaluation* **20** (1994) 337–357.
- [18] S. Christensen, Coloured Petri Nets Extended with Place Capacities, Test Arcs and Inhibitor Arcs, in: M. A. Marsan, Ed., *Application and Theory of Petri Nets 1993, Lecture Notes in Computer Science*, Bd. 691 (Springer Verlag, 1993) 186–205.
- [19] G. Ciardo, A. Blakemore, P. F. J. Chimento, J. K. Muppala, und K. Trivedi, Automated generation and analysis of Markov reward models using Stochastic Reward Nets, in: C. Meyer und R. J. Plemmons, Ed., *Linear Algebra, Markov Chains, and Queueing Models, IMA Volumes in Mathematics and its Applications*, Bd. 48 (Springer Verlag, 1992).
- [20] G. Ciardo, R. German, und C. Lindemann, A Characterization of the Stochastic Process Underlying a Stochastic Petri Net, *IEEE Transactions on Software Engineering* **20** (1994) 506–515.
- [21] D. Corbeel, J. C. Gentina, und C. Vercauter, Application of an extension of Petri nets to modelization of control and production processes, in: G. Rozenberg, Ed., *Advances in Petri Nets 1985, Lecture Notes in Computer Science*, Bd. 222 (1986) 163–180.
- [22] J. Couvillion, R. Freire, R. Johnson, W. D. Obal, M. A. Qureshi, M. Rai, und W. H. Sanders, Performability Modeling with UltraSAN, *IEEE Software* **8** (1991) 69–80.

- [23] J. M. Couvreur, S. Haddad, und J. F. Peyre, Generative families of positive invariants in Coloured nets sub-classes, in: G. Rozenberg, Ed., *Advances in Petri Nets 1993, Lecture Notes in Computer Science*, Bd. 674 (Springer Verlag, 1993).
- [24] K. Dalkowski, Modellierung und Bewertung einer Montagelinie mit speziellen Petrinetzen, Diplomarbeit, Technische Universität Berlin, September 1996.
- [25] C. Dutheillet und S. Haddad, Conflict Sets in Colored Petri Nets, in: *Proc. Int. Workshop on Petri Nets and Performance Models* (Toulouse, 1993) 76–85.
- [26] W. Eversheim, *Organisation in der Produktionstechnik, Studium und Praxis*, Bd. 3, Arbeitsvorbereitung (VDI Verlag, 1988).
- [27] K. Feldmann, W. Colombo, und C. Schnur, An Approach for Modeling, Analysis and Real-Time Control of Flexible Manufacturing Systems Using Petri Nets, in: *Proc. European Simulation Symposium* (Erlangen-Nürnberg, 1995) 661–665.
- [28] H. Genrich und K. Lautenbach, System modelling with high-level Petri nets, *Theoretical Computer Science* **13** (1981) 109–136.
- [29] J. C. Gentina und D. Corbeel, Coloured Adaptive Structured Petri-Net : A Tool for the Automatic Synthesis of Hierarchical Control of Flexible Manufacturing Systems, in: *Proc. IEEE Int. Conf. on Robotics and Automation* (1987) 1166–1173.
- [30] R. German, D. Logothetis, und K. S. Trivedi, Transient Analysis of Markov Regenerative Stochastic Petri Nets: A Comparison of Approaches, in: *Proc. 6th Int. Workshop on Petri Nets and Performance Models* (Durham, North Carolina, 1995) 103–112.
- [31] R. German und J. Mitzlaff, Transient Analysis of Deterministic and Stochastic Petri Nets with TimeNET, in: *Proc. Joint Conference 8th Int. Conf. on Modelling Techniques and Tools for Performance Evaluation, Lecture Notes in Computer Science*, Bd. 977 (Springer Verlag, 1995) 209–223.
- [32] R. German, *Analysis of Stochastic Petri Nets with Non-Exponentially Distributed Firing Times*, Dissertation, Technische Universität Berlin, 1994.
- [33] R. German, C. Kelling, A. Zimmermann, und G. Hommel, TimeNET – A Toolkit for Evaluating Non-Markovian Stochastic Petri Nets, *Performance Evaluation* **24** (1995) 69–87.
- [34] R. German, A. Zimmermann, C. Kelling, und G. Hommel, Modellierung und Bewertung von flexiblen Fertigungssystemen mit TimeNET, *CIM Management* (3) (Juni 1994) 24–27.
- [35] S. Haddad, Structural Analysis of Coloured Petri Nets, Tutorial auf der Int. Conf. on Petri Nets and Performance Models (Toulouse, 1993).
- [36] I. Hatono, K. Yamagata, und H. Tamura, Modeling and On-Line Scheduling of Flexible Manufacturing Systems, *IEEE Transactions on Software Engineering* **17** (1991) 126–132.

- [37] A. Heindl und R. German, A Fourth Order Algorithm with Automatic Stepsize Control for the Transient Analysis of DSPNs, Technischer Bericht des Fachbereichs Informatik 35, Technische Universität Berlin, 1996.
- [38] K. Jensen und G. Rozenberg, Ed., *High-level Petri nets: theory and applications* (Springer Verlag, 1991).
- [39] K. Jensen, Coloured Petri Nets and the Invariant-Method, *Theoretical Computer Science* **14** (1981) 317–336.
- [40] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, EATCS Monographs on Theoretical Computer Science (Springer Verlag, 1992).
- [41] E. Kasturia, F. DiCesare, und A. A. Desrochers, Real Time Control of Multilevel Manufacturing Systems Using Colored Petri Nets, in: *Proc. Int. Conf. on Robotics and Automation* (1988) 1114–1119.
- [42] C. Kelling, *Simulationsverfahren für zeiterweiterte Petri-Netze*, Dissertation, Technische Universität Berlin, 1995, Advances in Simulation, SCS International.
- [43] K. Koischwitz, Entwurf und Implementierung einer parametrisierbaren Benutzungsoberfläche für hierarchische Netzmodelle (agnes - ein Generisches Netz-Editier-System), Diplomarbeit, Technische Universität Berlin, Oktober 1996.
- [44] K. Kurbel, *Produktionsplanung und -steuerung, Handbuch der Informatik*, Bd. 13.2 (Oldenbourg, 1993).
- [45] A. M. Law und W. D. Kelton, *Simulation Modeling and Analysis* (McGraw-Hill, 1991).
- [46] D. Y. Lee und F. DiCesare, FMS Scheduling Using Petri Nets and Heuristic Search, in: *Proc. Int. Conf. on Robotics and Automation* (Nice, 1992) 1057–1062.
- [47] C. Lin und D. C. Marinescu, On Stochastic High-Level Petri Nets, in: *Proc. 2nd Int. Workshop on Petri Nets and Performance Models* (Madison, Wisconsin, 1987) 34–43.
- [48] C. Lin und D. C. Marinescu, Stochastic High-level Petri Nets and Applications, *IEEE Transactions on Computers* **37** (1988) 815–825.
- [49] C. Lindemann, G. Ciardo, R. German, und G. Hommel, Performability Modeling of an Automated Manufacturing System with Deterministic and Stochastic Petri Nets, in: *Proc. IEEE Int. Conf. on Robotics and Automation* (Atlanta, Georgia, 1993) 576–581.
- [50] C. Lindemann, *Stochastic Modeling using DSPNexpress* (Oldenbourg, 1994).
- [51] C. Lindemann und A. Zimmermann, An Adaptive Algorithm for the Efficient Generation of the Tangible Reachability Graph of a Stochastic Petri Net, Technischer Bericht des Fachbereichs Informatik 8, Technische Universität Berlin, 1994.
- [52] J. Martínez, P. Muro, und M. Silva, Modeling, Validation and Software Implementation of Production Systems Using High Level Petri Nets, in: *Proc. Int. Conf. on Robotics and Automation* (Raleigh, North Carolina, 1987) 1180–1185.

- [53] J. Martínez und M. Silva, A language for the description of concurrent systems modelled by coloured Petri nets: Application to the control of flexible manufacturing systems, in: *Proc. of the 1984 IEEE Workshop on Languages for Automation* (New Orleans, 1984) 72–77.
- [54] J. Martínez und M. Silva, A simple and fast algorithm to obtain all invariants of a generalised Petri net, in: W. R. C. Girault, Ed., *Second Eur. Workshop on Application and Theory of Petri Nets, Informatik-Fachberichte*, Bd. 52 (Springer Verlag, 1982) 301–310.
- [55] K. Mertins und M. Rabe, Flexible Abbildung von Produktionseinrichtungen und Steuerung mit dem Planungssystem MOSYS, in: *Fortschritte der Simulationstechnik Band 4* (Vieweg Verlag, 1991) 386–390.
- [56] M. K. Molloy, Performance analysis using stochastic Petri nets, *IEEE Transactions on Computers* **31** (9) (1982) 913–917.
- [57] M. K. Molloy, Discrete Time Stochastic Petri Nets, *IEEE Transactions on Software Engineering* **11** (4) (April 1985) 417–423.
- [58] J.-Y. Morel, M. Bourcerie, und J.-L. Ferrier, Scheduling with Petri Networks, *Petri Net Newsletter* (44) 15–19.
- [59] T. Murata, Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE* **77** (4) (1989) 541–580.
- [60] Y. Narahari und N. Viswanadham, On the Invariants of Coloured Petri Nets, in: G. Rozenberg, Ed., *Advances in Petri Nets 1985, Lecture Notes in Computer Science*, Bd. 222 (Springer Verlag, 1986) 330–345.
- [61] C. Petri, *Kommunikation mit Automaten*, Dissertation, Schriften des Institutes für Instrumentelle Mathematik, Bonn, 1962.
- [62] W. Reisig, *Petri nets* (Springer Verlag Berlin, 1985).
- [63] E. Schnieder, Grundlagen, Perspektiven und Visionen einer Entwurfsmethodik für Automatisierungssysteme, in: E. Schnieder, Ed., *4. Fachtagung Entwurf komplexer Automatisierungssysteme* (Braunschweig, 1995) 1–20.
- [64] E. Schnieder, *Petri-Netze in der Automatisierungstechnik* (Oldenbourg Verlag, 1992).
- [65] A. Schömig und O. Gühr, Leistungsuntersuchung von Schedulingverfahren in Produktionssystemen, in: *Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (MMB)* (1993) 74–85.
- [66] G. Seliger, Skript der Vorlesung Produktionsplanung, Technische Universität Berlin, 1990.
- [67] B. Shansagimow und E. Schnieder, Analyse und Auswahl optimaler Varianten von Automatisierungssystemen mit Hilfe von Petrinetzdarstellungen, *Automatisierungstechnik – at* **40** (6) (1992) 228–234.

- [68] J. Sifakis, Structural properties of Petri nets, in: J. Winkowski, Ed., *Mathematical Foundations of Computer Science* (Springer Verlag, 1978) 474–483.
- [69] M. Silva und E. Teruel, Petri Nets for the Design and Operation of Manufacturing Systems, in: *Proc. 5th Int. Conf. on Computer Integrated Manufacturing and Automation Technology (CIMAT '96)* (Lille, 1996) .
- [70] M. Silva und R. Valette, Petri Nets and Flexible Manufacturing, in: G. Rozenberg, Ed., *Advances in Petri Nets 1989, Lecture Notes in Computer Science*, Bd. 424 (Springer Verlag, 1989) 374–417.
- [71] G. Spur, K. Mertins, B. Wieneke-Toutaoui, und M. Rabe, MOSYS – a planning system for manufacturing and assembly planning, in: *Simulation in Manufacturing (SIM5)* (Bedford, 1989, IFS) .
- [72] P. Starke, *Analyse von Petri-Netz-Modellen* (Teubner Verlag, Stuttgart, 1990).
- [73] N. Treves, A Comparative Study of Different Techniques for Semi-Flows Computation in Place/Transition Nets, in: G. Rozenberg, Ed., *Advances in Petri Nets 1989, Lecture Notes in Computer Science*, Bd. 424 (Springer Verlag, 1990) 434–452.
- [74] K. S. Trivedi und V. G. Kulkarni, FSPNs: Fluid Stochastic Petri Nets, in: *Proc. 14th Int. Conf. on Application and Theory of Petri Nets* (Chicago, Illinois, USA, 1990) 24–31.
- [75] J. L. Villaroel, J. Martínez, und M. Silva, GRAMAN: A Graphic System for Manufacturing System Design, in: S. Tzafestas, Ed., *IMACS Symposium on System Modelling and Simulation* (1989, Elsevier Science Publ.) 311–316.
- [76] N. Viswanadham und Y. Narahari, Coloured Petri Net Models for Automated Manufacturing Systems, in: *Proc. Int. Conf. on Robotics and Automation* (Raleigh, North Carolina, 1987) 1985–1990.
- [77] N. Viswanadham und Y. Narahari, *Performance Modeling of Automated Manufacturing Systems* (Prentice-Hall Inc., 1992).
- [78] T. Watanabe und M. Yamauchi, New Priority-Lists for Scheduling in Timed Petri Nets, in: M. A. Marsan, Ed., *Application and Theory of Petri Nets 1993, Lecture Notes in Computer Science*, Bd. 691 (Springer Verlag, 1993).
- [79] J. F. Watson und A. A. Desrochers, Applying Generalized Stochastic Petri Nets to Manufacturing Systems containing Non-exponential Transition Functions, *IEEE Transactions on System, Man, and Cybernetics* **21** (1991) 1008–1017.
- [80] H. P. Wiendahl, *Belastungsorientierte Fertigungssteuerung* (Carl Hanser, 1987).
- [81] H. P. Wiendahl, *Analyse und Neuordnung der Fabrik* (Springer Verlag, 1991).
- [82] A. Zenie, Colored Stochastic Petri Nets, in: *Proc. 1st Int. Workshop on Petri Nets and Performance Models* (1985) 262–271.

- [83] M. Zhou und F. DiCesare, A Top-down Approach to Systematic Synthesis of Petri Net Models for Manufacturing Systems, in: *Proc. Int. Conf. on Robotics and Automation* (1989) 534–539.
- [84] M. Zhou und F. DiCesare, A Petri Net Design Method for Automated Manufacturing Systems with Shared Resources, in: *Proc. Int. Conf. on Robotics and Automation* (1990) 526–531.
- [85] M. Zhou, K. McDermott, und P. A. Patel, Petri Net Synthesis and Analysis of a Flexible Manufacturing System Cell, *IEEE Transactions on System, Man, and Cybernetics* **23** (2) (1993) 523–531.
- [86] A. Zimmermann, S. Bode, und G. Hommel, Performance and Dependability Evaluation of Manufacturing Systems Using Petri Nets, in: *1st Workshop on Manufacturing Systems and Petri Nets, 17th Int. Conf. on Application and Theory of Petri Nets* (Osaka, Japan, 1996) 235–250.
- [87] A. Zimmermann, K. Dalkowski, und G. Hommel, A Case Study In Modeling And Performance Evaluation Of Manufacturing Systems Using Colored Petri Nets, in: *Proc. of the 8th European Simulation Symposium* (Genoa, 1996) 282–286.
- [88] A. Zimmermann, A Modeling Method for Flexible Manufacturing Systems based on Colored Petri Nets, in: *Proc. Int. Workshop on New Directions of Control and Manufacturing* (Hong Kong, 1994) 147–154.
- [89] A. Zimmermann und S. Bode, Modellierung von Fertigungssystemen und Arbeitsplänen mit farbigen Petri-Netzen, in: E. Schnieder, Ed., *4. Fachtagung Entwurf komplexer Automatisierungssysteme* (Braunschweig, 1995) 249–262.
- [90] R. Z. Zurawski und T. S. Dillon, Systematic Construction of Functional Abstractions of Petri Net Models of Typical Components of Flexible Manufacturing Systems, in: *Proc. 5th Int. Workshop on Petri Nets and Performance Models* (Toulouse, 1993) 248–257.