

In: E. Schnieder (Ed.), 6. Fachtagung Entwicklung und Betrieb komplexer Automatisierungssysteme (EKA99), Braunschweig 1999, S. 31–48.

Fertigungssysteme Modellieren, Bewerten und Steuern mit TimeNET_{MS}

Armin Zimmermann, Jörn Freiheit und Günter Hommel Institut für Technische Informatik
Technische Universität Berlin
Franklinstr. 28/29, Sekr. FR 2-2
10587 Berlin
Tel.: (30) 314 - 73 112
Fax.: (30) 314 - 21 116
E-Mail: {azi, freiheit, hommel}@cs.tu-berlin.de

Abstract: Moderne Fertigungssysteme erfordern in den verschiedenen Phasen ihres Lebenszyklus den Einsatz modellbasierter Softwarewerkzeuge. Bereits während der Planung müssen Verhalten und Leistungsfähigkeit eines Fertigungsprozesses möglichst genau bestimmt werden können. Verschiedene Steuerstrategien können anhand eines Modells automatisch auf Verklemmungsfreiheit sowie ihre Auswirkungen auf den Durchsatz oder andere Kennzahlen hin untersucht werden. Dieser Beitrag beschreibt TimeNET_{MS}, ein Werkzeug zur durchgängigen Modellierung, Validierung, Leistungsbewertung und Steuerung von Fertigungssystemen. Diese verschiedenen Aufgaben werden auf der Grundlage eines gemeinsamen Petri-Netz-Modells durchgeführt, so daß keine Konvertierung zwischen verschiedenen Beschreibungsmitteln notwendig ist.

Stichworte: Fertigungssysteme, Petri-Netze, Leistungsbewertung, Steuerung, Werkzeug

1 Einleitung

Durch die Entwicklung auf dem Gebiet der Produktionstechnik werden immer komplexere Fertigungssysteme entwickelt und eingesetzt. Die erforderlichen Anlaufinvestitionen steigen, bevor ein Nutzen zu erzielen ist. Dem Entwicklungsprozeß kommt demnach eine hohe Bedeutung zu, auf der anderen Seite wird er komplexer. Aussagen über das spätere Verhalten eines geplanten Systems sind für fundierte Entscheidungen notwendig. Neben Fragen nach der korrekten Funktionsweise ist meist die Leistungsfähigkeit und Zuverlässigkeit interessant. Ein wichtiges Hilfsmittel zur Lösung der genannten Probleme stellt die Modellierung und darauf basierende Untersuchung der geplanten Systeme dar. Es kann darüber hinaus verwendet werden, um eine Steuerung des geplanten Systems zu entwerfen und zu überprüfen. In diesem Beitrag wird kein automatischer Steuerungsentwurf durchgeführt. Statt dessen ist das Werkzeug als ein Hilfsmittel zur Erprobung von Steuerstrategien bzw. zur Nachbildung des Streckenverhaltens zu

verstehen, an dem eine entworfene Steuerstrategie erprobt werden kann. Von großer Wichtigkeit für die Akzeptanz eines Softwarewerkzeugs und der zugrundeliegenden Methoden ist die Wiederverwendbarkeit der Modelle in verschiedenen Entwurfs- und Betriebsphasen.

Dieser Beitrag stellt das Softwarewerkzeug TimeNET_{MS} (*timed net evaluation tool for manufacturing systems*) vor, mit dem ein Schritt hin zur Lösung der oben genannten Probleme unternommen wird. In ihm sind neben einer aktuellen Erweiterung um die Steuerung von Fertigungssystemen die in [Z98] vorgeschlagenen Methoden implementiert. Die Beschreibung der Fertigungsprozesse erfolgt dabei mit speziellen zeiterweiterten farbigen Petri-Netzen.

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 [M89]. Durch die lokale Beschreibung von Aktivitäten und ihres Einflusses auf den Zustand sind sie für Verfeinerung und modulare Konstruktion eines Modells geeignet. Eine klassische Anwendung sind Fertigungssysteme von der Produktionsplanung bis hin zur Fertigungssteuerung [ST96].

Ziel der hier beschriebenen Forschungsarbeiten ist eine in einem Werkzeug realisierte Methode, die den Entwurf und den Betrieb eines Fertigungsprozesses durchgängig ohne Wechsel des Beschreibungsmittels modellgestützt erleichtert. Informationsverluste und Fehler bei der Übertragung zwischen verschiedenen Modellformen in den Entwurfs- und Betriebsphasen des Fertigungsprozesses werden so vermieden. Abb. 1 zeigt vereinfacht die Abfolge der dazu gehörenden Schritte. 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 [Z98], die im Abschnitt 3 beschrieben wird. Damit ist der Modellierer in der Lage, getrennte Modelle der Struktur und der Arbeitspläne zu entwerfen. Dieser Aspekt ist bei der Modellierung mit Petri-Netzen bisher nicht ausreichend beachtet worden. Das Strukturmodell entspricht dabei weitgehend dem realen Aufbau des Fertigungssystems, was die Verständlichkeit erleichtert. Um die Entwicklung großer Modelle zu erleichtern, wird eine Bibliothek von Teilmodellen eingesetzt.

Alternativ zur Modellierung mit speziellen Petri-Netzen ist die Modellierung mit Funktionssymbolen möglich [ZKH98, Z98]. Hier wird jedoch nicht auf diesen Aspekt eingegangen.

Aus den Modellen der Struktur und der Arbeitspläne wird automatisch ein Gesamtmodell erzeugt. Dieses stellt eine Erweiterung des Strukturmodells um Informationen aus den verschiedenen Arbeitsplänen dar. Es entsteht ein das gesamte Verhalten modellierendes Petri-Netz, welches im Anschluß mit Hilfe der interaktiven Simulation (Markenspiel), einer automatisch ablaufenden Animation, strukturellen Untersuchungsmethoden und anhand des Erreichbarkeitsgraphen auf seine korrekte Funktionsweise überprüft werden kann. Mit Hilfe von Simulation

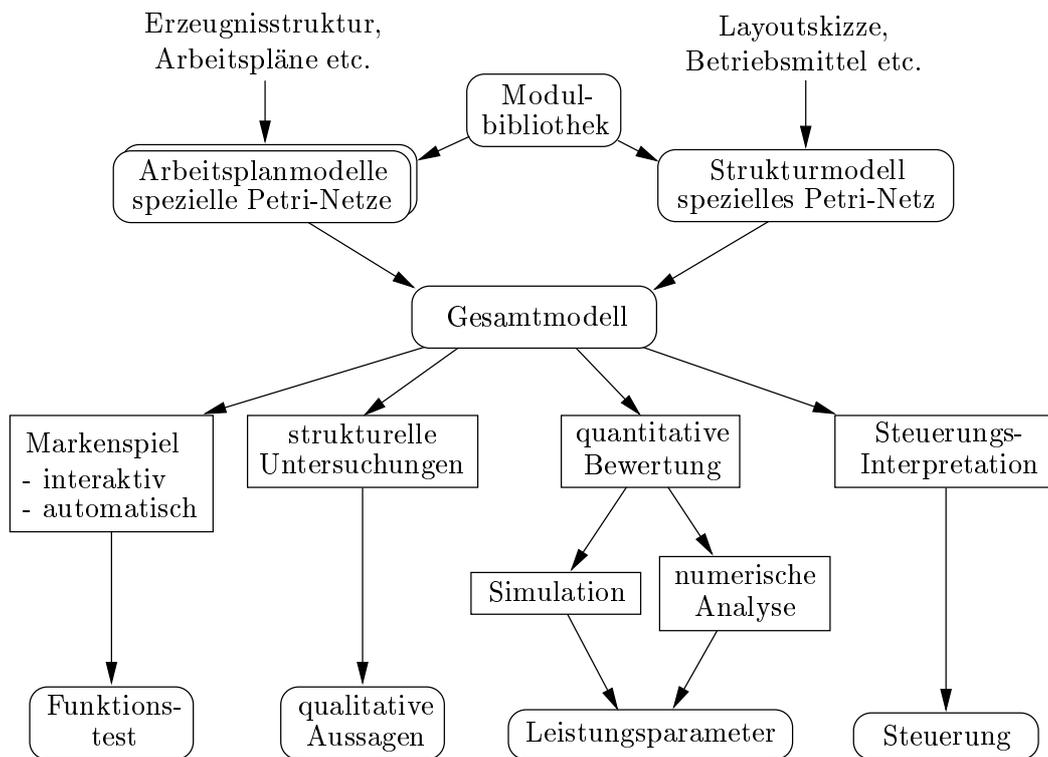


Abbildung 1: Überblick über die Modellierungs- und Bewertungsmethode.

oder numerischen Analyseverfahren werden Leistungsparameter des geplanten Fertigungssystems bestimmt. Dadurch ist eine Leistungsbewertung unter Einbeziehung der vorgesehenen Steuerstrategien sowie des Ausfall- und Reparaturverhaltens möglich. Durch den Vergleich verschiedener Varianten kann eine Optimierung vorgenommen werden. Dieser Schritt wird im Abschnitt 4 beschrieben. Wird das in den vorangegangenen Schritten entstandene Modell um Informationen über die Anbindung beispielsweise von Motoren und Sensoren erweitert, kann es ohne weitere Veränderungen zur Steuerung der modellierten Fertigungsanlage eingesetzt werden. Abschnitt 5 geht auf die Einzelheiten ein.

Die entwickelten Verfahren wurden in einem Softwarewerkzeug als Erweiterung von TimeNET [GKZH95] realisiert. Die eingesetzte Benutzungsoberfläche muß für verschiedene Modellarten verwendbar sein. Darum wurde eine parametrisierbare Benutzungsoberfläche für hierarchische Graphen entwickelt, die im Abschnitt 3.3 vorgestellt wird. Eine Gegenüberstellung der hier vorgeschlagenen Methode und Beschreibungsmittel mit anderen, ähnlichen Arbeiten kann [Z98] entnommen werden.

2 Ein Anwendungsbeispiel

Abb. 2 zeigt das im folgenden verwandte Beispiel-Fertigungssystem, eine Modellfertigungsanlage aus Fischertechnik-Elementen. Der Austausch von Werkstücken findet über das Hochregallager statt. Der Portalroboter kann mit einem Greifer Werkstücke vom Übergabepplatz zum Förderband transportieren. Von dort aus werden Werkstücke über die drei Förderbänder zu

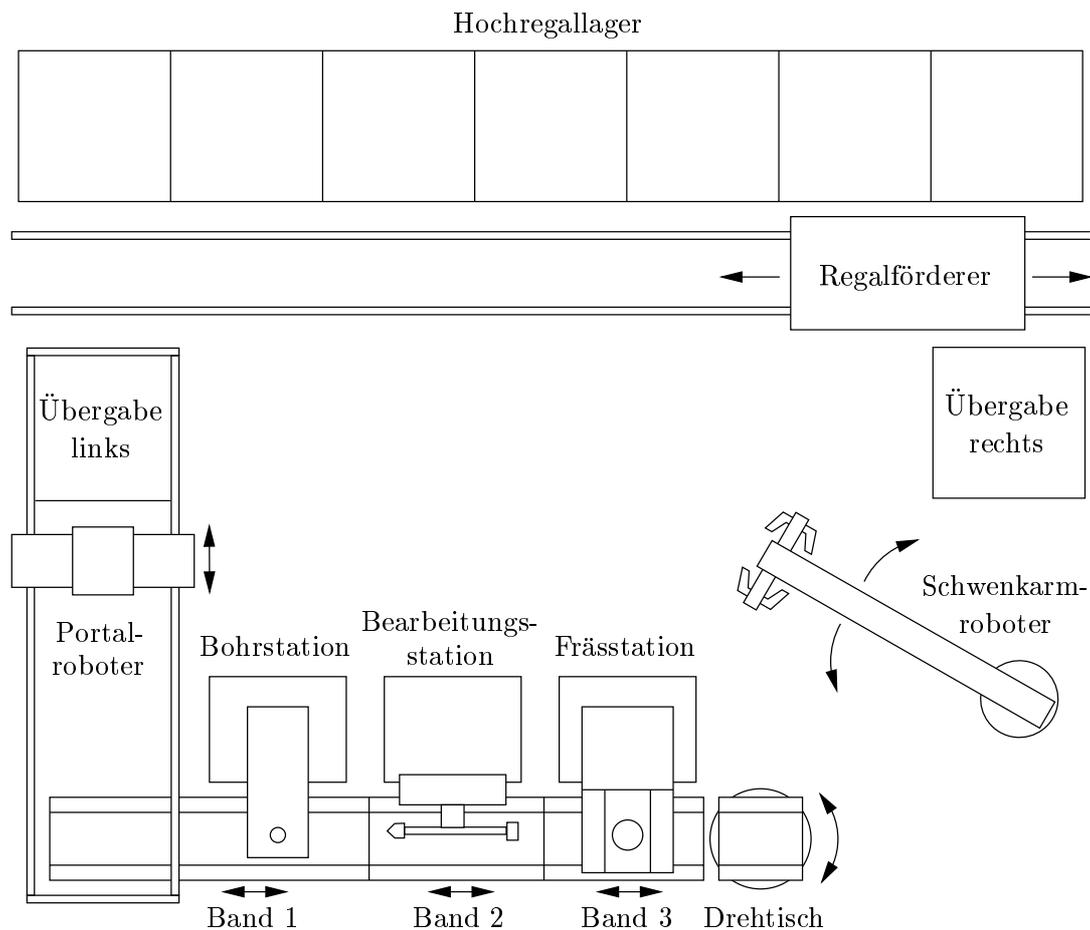


Abbildung 2: Aufbau des Fertigungsanlagenmodells.

den Maschinen transportiert. Als Bearbeitungsstationen sind eine Bohrmaschine, eine Bearbeitungsstation mit dreifach-Werkzeugwechsler sowie eine Fräse vorhanden. Der Schwinkarmroboter befördert die Werkstücke zum Übergabepplatz, von wo sie durch den Regalförderer aufgenommen werden können.

3 Modellierung von Fertigungssystemen mit Petri-Netzen

Klassische Petri-Netze mit nicht unterscheidbaren Marken sind für die Modellierung nichttrivialer Fertigungssysteme kaum verwendbar. Stellen und Transitionen müssen entsprechend den auftretenden Produktarten vervielfältigt werden. Dies führt zu großen und unübersichtlichen Modellen. Farbige Petri-Netze [J92] bieten die Möglichkeit, den Marken Attribute zuzuweisen und dadurch kompaktere Modelle zu erstellen. Außerdem sind größere Modelle hierarchisch strukturierbar. Leider ist jedoch die programmiersprachenartige Definition von Typen, Variablen und Funktionen notwendig. Diese und weitere textuelle Elemente erschweren die Verständlichkeit eines Modells.

Darum wurden für das Anwendungsgebiet der Fertigungssysteme spezielle farbige Petri-Netze eingeführt [Z98], in denen zwei Markentypen 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. Für die definierte Netzklasse werden die erlaubten Markentypen der farbigen Petri-Netze eingeschränkt. Es werden nur zwei Arten von Marken zugelassen: ungefärbte, ununterscheidbare Marken (Elementarmarken) für logische Zustände und die Farbe „Produkt“. Produktmarken entsprechen z.B. einem Werkstück oder einem Werkstückträger im Fertigungssystem.

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. 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 deshalb als mit dem Strukturmodell *assoziierte Petri-Netze* bezeichnet.

In den folgenden Abschnitten wird die Modellfertigungsanlage mit den genannten speziellen farbigen Petri-Netzen abgebildet.

3.1 Modellierung der Struktur

Abb. 3 zeigt das Petri-Netz-Modell der Struktur des Anwendungsbeispiels aus Abschnitt 2 auf der obersten Hierarchiestufe. Bei der Modellierung wurde nach dem Prinzip verfahren, Aufenthaltsorte von Werkstücken mit Stellen und Maschinen sowie Transporteinrichtungen mit Transitionen abzubilden. Durch diese klare Beziehung zwischen Realität und Modell ist letzteres leicht verständlich. Transitionen mit dick gezeichnetem oberem und unterem Rand sind Vertretertransitionen, d.h. sie sind mit einem Petri-Netz auf der nächstniedrigeren Hierarchiestufe verfeinert. Im Beispiel wird so im Interesse der Übersichtlichkeit das innere Verhalten und der Zustand der Maschinen und Transporteinrichtungen versteckt. Die verfeinerten Modelle beschreiben die genauen Abläufe dieser aktiven Betriebsmittel, die eng mit der Steuerung zusammenhängt. Abschnitt 5 enthält in Abb. 7 eine Darstellung des verfeinerten Modells des Schwenkarmroboters, während hier und im folgenden lediglich auf die oberste Hierarchiestufe eingegangen wird. Das gesamte Strukturmodell des Beispiels besteht aus 12 einzelnen sogenannten Seiten. In einem anderen, größeren Anwendungsbeispiel [ZDH96] wurden die Unterseiten außerdem dazu verwendet, das genaue Ausfall- und Reparaturverhalten abzubilden.

Jede Stelle im gezeigten Modell entspricht einem möglichen Aufenthaltsort eines Werkstücks. Im Modell werden vereinfachend nur zwei Palettenplätze des Hochregallagers (**Fach1** und **Fach2**) betrachtet. An Stellen stehende Zahlen in eckigen Klammern geben Kapazitäten an. Durch die Bearbeitung eines Werkstückes ändert sich nur der Bearbeitungszustand, nicht jedoch der Ort. Deshalb sind häufig (wie im Bild zu sehen) Pfeile zwischen Stellen und Transitionen in beiden Richtungen vorhanden, die der Einfachheit halber als Doppelpfeil gezeichnet werden.

Die Transition **AnAb liefern** mit modelliert den Austausch von angelieferten bzw. fertiggestellten Werkstücken mit der Außenwelt, der über das Hochregallager stattfindet. Ihre grafische Darstellung symbolisiert, daß es sich um eine nicht verfeinerte Transition mit fester, also deterministischer Schaltzeit handelt. Transportvorgänge zwischen den Maschinen sind immer nur durch

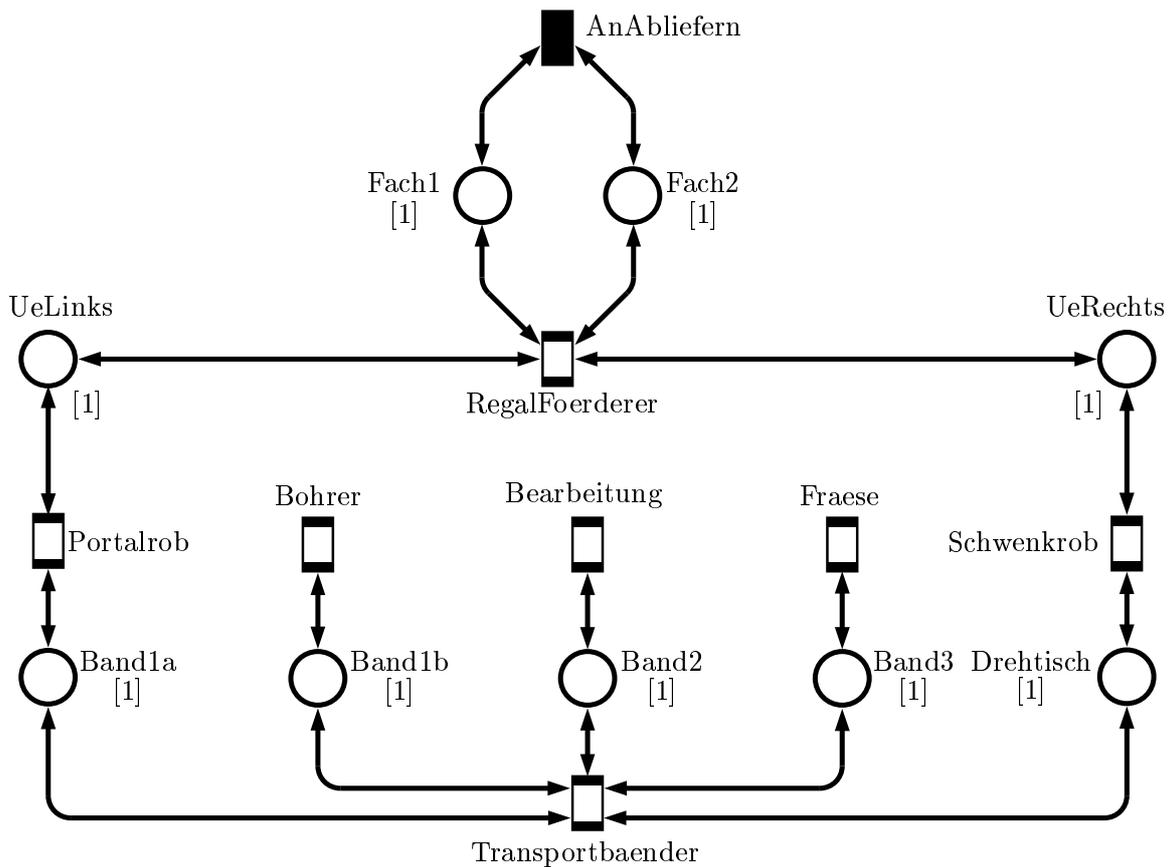


Abbildung 3: Strukturmodell auf der obersten Hierarchiestufe.

das Zusammenspiel zweier Transportbänder zu erreichen. Aus diesem Grund wird das Verhalten aller drei Transportbänder gemeinsam in einer Vertretertransition **Transportbaender** zusammengefaßt. Die restlichen Transitionen beschreiben jeweils eine Maschine oder Transporteinrichtung entsprechend des in Abb. 2 gezeigten Aufbaus.

3.2 Modellierung der Arbeitspläne

Neben dem im vorangegangenen Abschnitt beschriebenen Strukturmodell werden zur Beschreibung eines Produktionsprozesses die Arbeitspläne aller zu fertigenden Werkstücke benötigt. Innerhalb der hier verwendeten Modellierungsmethode werden dafür ebenfalls spezielle farbige Petri-Netze eingesetzt, so daß nur ein Beschreibungsmittel notwendig ist. Für eines der beiden mit der Modellfertigungsanlage produzierten Werkstücke — hier A genannt — ist das entsprechende Modell in Abb. 4 dargestellt. Wie das entsprechende Strukturmodell ist es hierarchisch aus Seiten aufgebaut.

An den Kanten im Arbeitsplanmodell ist der Typ der transportierten oder bearbeiteten Teile (modelliert durch Marken) verzeichnet. Fertigungsalternativen wurden im Beispiel nicht benötigt, können aber im Modell problemlos abgebildet werden. Dabei ist es möglich, zur Abbildung bestimmter Materialflußverteilungen bzw. Transportstrategien einen bestimmten Prozentsatz der Werkstücke für eine Alternative vorzusehen oder das Betreten von Alternativen

vom aktuellen Zustand des Fertigungssystems abhängig zu gestalten.

Werkstücke werden auf Paletten transportiert. Dies wird nachgebildet, indem beim Schalten der Transition **AnAbliefen** eine Montageoperation mit der Palette vorgenommen wird. Auch für die leeren Paletten ist ein eigenes Arbeitsplanmodell vorhanden, in dem beispielsweise die Transportstrategie des Regalförderers für leere Paletten abgebildet wird.

3.3 Eine generische Benutzungsoberfläche

Innerhalb der hier beschriebenen Methodik werden verschiedene grafische Beschreibungsmittel für Fertigungssysteme eingesetzt. Ein benutzungsfreundliches Softwarewerkzeug muß den Umgang mit allen diesen Modellarten unterstützen. Darum wurde eine generische Benutzungsoberfläche für hierarchische grafische Modelle realisiert. Jeder Modelltyp wird als eine *Netzklasse* bezeichnet.

Die Definition einer Netzklasse soll ohne Programmieraufwand und Kenntnisse der Implementierung der Benutzungsoberfläche möglich sein. Sie erfolgt darum in einer Textdatei. In einer

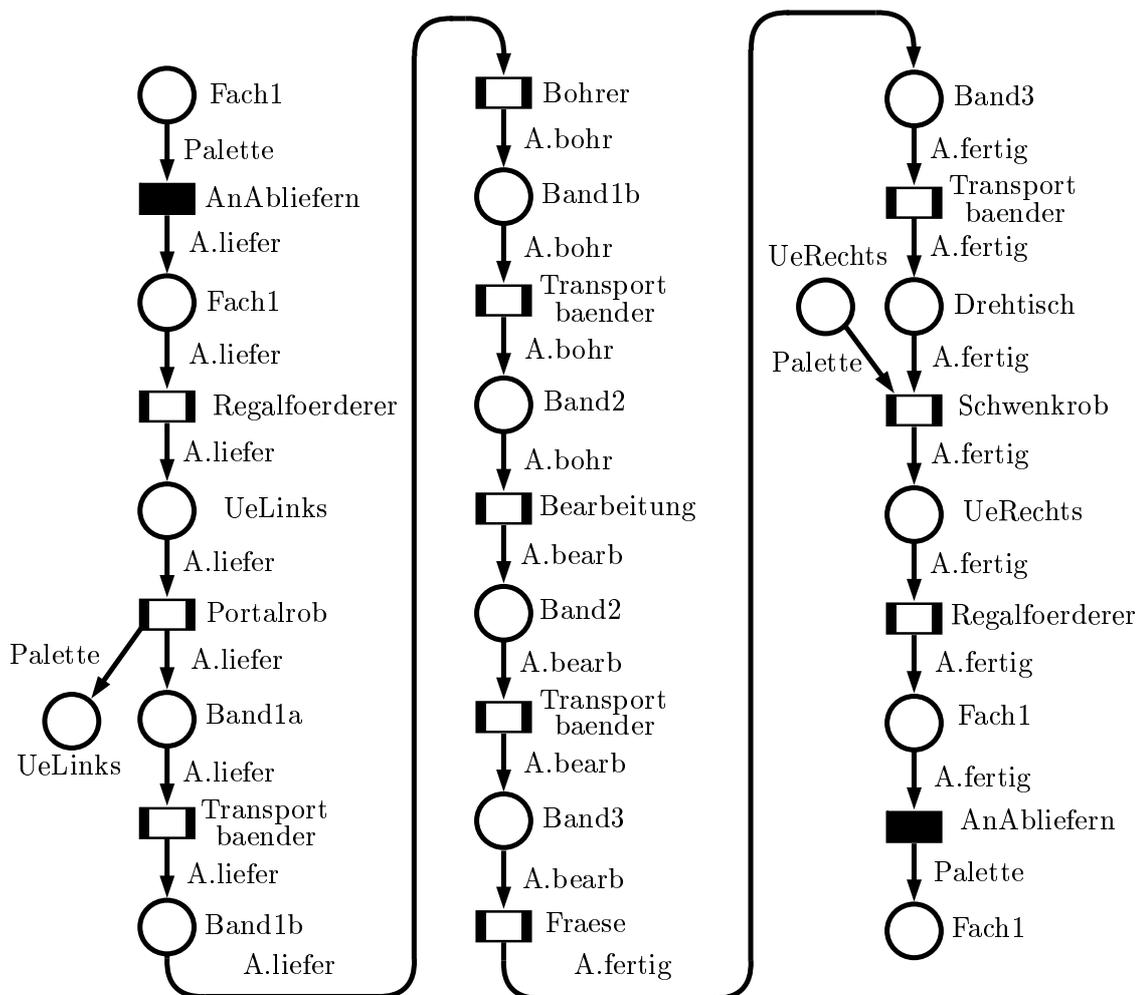


Abbildung 4: Arbeitsplanmodell für Werkstück A auf der obersten Hierarchiestufe.

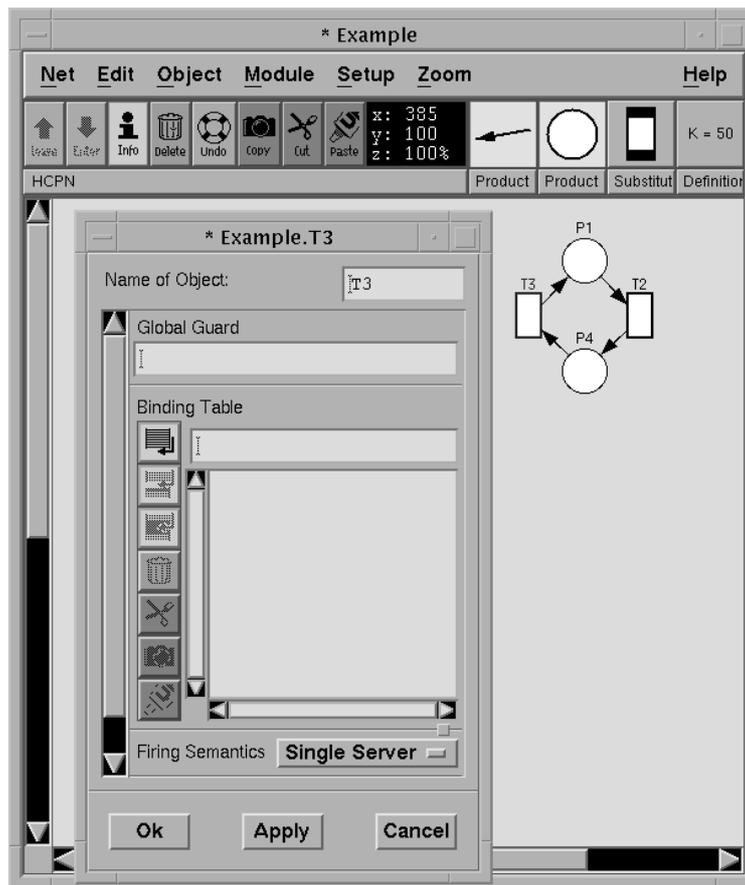


Abbildung 5: Bild der Benutzungsoberfläche.

solchen Netzklassendatei werden 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.

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. Der einfache Grundaufbau der grafischen Beschreibung mit vektororientierten Elementen erleichtert das Skalieren, Rotieren sowie Exportieren in andere Formate wie *PostScript* oder *xfig*.

Die Abb. 5 zeigt eine beispielhafte Bildschirmansicht der Benutzungsoberfläche Agnes. Die 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 wie z.B. Animation oder Simulation realisiert werden.

3.4 Überprüfung des Modells

Nach oder möglichst schon während der Eingabe eines Modells muß der Modellierer in der Lage sein, das Modell zu überprüfen. 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, indem die schaltfähigen Transitionen blinken. Der Modellierer wählt aus, welche davon ausgeführt werden soll, und kann so das Verhalten des Modells ausprobieren und mögliche Fehler finden. Alternativ ist eine Animation möglich, bei der eine langsame Simulation des Netzverhaltens mit gleichzeitiger Darstellung des Modellzustands durchgeführt wird.

Anschließend können grundsätzliche Fragen über die Funktionsweise des modellierten Systems beantwortet werden, die meist mit Ja oder Nein zu beantworten sind. 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.

TimeNET_{MS} bietet die automatische Berechnung der Stellen-Invarianten und Konfliktmengen des Modells an. Eine Erweiterung um Transitions-Invarianten, Traps, Siphons sowie notwendige Bedingungen für Lebendigkeit und Beschränktheit wird derzeit implementiert [HFZ98].

Bei der Eingabe des Petri-Netz-Modells der Modellfertigungsanlage wurden zunächst keinerlei Einschränkungen der Beladung leerer Paletten mit neuen Werkstücken im Hochregallager gemacht. Erste Untersuchungen des Erreichbarkeitsgraphen des Modells zeigten schnell, daß eine Verklemmung eintreten kann, wenn alle Plätze auf den Förderbändern mit Werkstücken belegt sind. Dies geschieht, wenn die letzte leere Palette mit einem unbearbeiteten Werkstück belegt wird, so daß keine Palette mehr für das Abholen fertiggestellter Produkte von der rechten Übergabestation vorhanden ist. Daraufhin wurde die Beladestrategie dahingehend geändert, daß nur dann die letzte leere Palette beladen wird, wenn es noch freie Plätze auf den Förderbändern gibt. Auf diese Weise war ohne zeitraubende Tests an der realen Anlage der Entwurf einer verklemmungsfreien Strategie möglich.

4 Leistungsbewertung

Im Anschluß kann eine Leistungsbewertung des modellierten Fertigungssystems erfolgen. 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 folgenden Abschnitten beschrieben.

Mit dem Werkzeug TimeNET_{MS} lassen sich in der derzeitigen Ausbaustufe Kenngrößen ermit-

eln, die sich aus der Wahrscheinlichkeitsverteilung der Systemzustände im eingeschwungenen Zustand berechnen lassen.

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. Die Wahrscheinlichkeit dafür, daß eine Maschine ausgefallen ist, kann über die Markierung der Stelle berechnet werden, die den Fehlerzustand der Maschine modelliert. 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. Für das Modellfertigungssystem soll der Gesamtdurchsatz untersucht werden. Dafür wurde ein Leistungsparameter definiert, der den Durchsatz einer Transition berechnet, die jedes Produkt A und B auf ihrem Weg durch das Fertigungssystem genau einmal passieren. Die gewählte Transition befindet sich auf der zur Vertretertransition `Regalfoerderer` gehörenden Unterseite, da der Durchsatz einer Vertretertransition nicht definiert ist.

Zur quantitativen Analysierbarkeit ist die Verwendung zeitbehafteter Vorgänge im Modell notwendig. Dabei wird auf Erweiterungen der stochastischen Petri-Netze aufgebaut [G94], 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 in `TimeNETMS` beide Untersuchungsmethoden zur Verfügung gestellt.

4.1 Analyse

Dieser Abschnitt beschreibt, wie aus einem speziellen Petri-Netz-Modell Leistungsparameter mit numerischen Verfahren gewonnen werden können. Dabei wird auf eine in [G94] entwickelte numerische Analysetechnik für erweiterte deterministische und stochastische Petri-Netze (*eDSPNs*) 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.

Für die numerische Analyse wird vorausgesetzt, daß der dem Modell zugrundeliegende Erreichbarkeitsgraph endlich ist und keine absorbierenden Zustände enthält. Außerdem darf in jeder Markierung maximal eine Transition mit nicht exponentiell verteilter Schaltzeit aktiviert sein. Die den Transitionen zuordenbaren Schaltzeitverteilungen können aus der Klasse der *exponentialen Verteilungen* ausgewählt werden. Die meisten gebräuchlichen Verteilungsfunktionen sind Spezialfälle dieser Klasse. Unter diesen Voraussetzungen ist der dem Modell zugrundeliegende stochastische Prozeß vom Typ *semi-regenerativ* [CGL94]. 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. Aus den stationären Zustandswahrscheinlichkeiten werden dann die Ergebnisse der Leistungsparameter berechnet und in der Benutzungsoberfläche angezeigt.

4.2 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. Letzteres war für die Modellfertigungsanlage der Fall, da für die Bearbeitungs- und Transportzeiten deterministische Schaltzeiten gewählt wurden. Dann können die angegebenen Leistungsparameter simulativ berechnet werden. Dieser Abschnitt erläutert die dabei verwendeten Techniken, die in [K95] für farblose Petri-Netze vorgestellt wurden und für TimeNET_{MS} entsprechend angepaßt wurden.

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.

Neben dem Mittelwert, der den tatsächlichen Wert des interessierenden Leistungsparameters 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. In TimeNET_{MS} wird die in [K95] beschriebene Methode der Spektralanalyse der Varianz verwendet. Sie setzt keine unabhängigen Einzelwerte voraus, sondern nur einen schwach stationären Prozeß.

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. Hier bietet sich eine Parallelisierung der Simulation

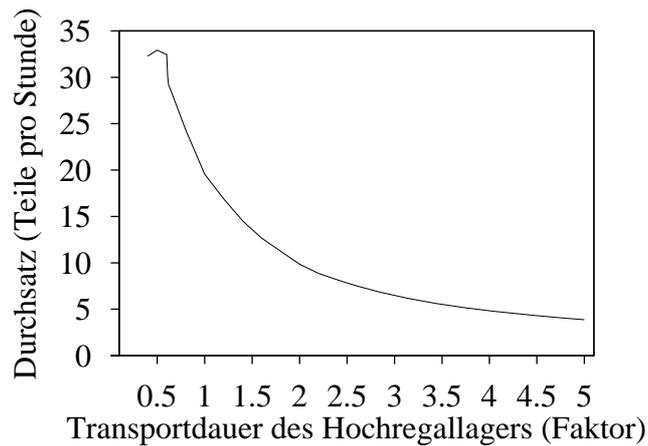


Abbildung 6: Durchsatz gegenüber Geschwindigkeit des Förderzeugs.

an. In TimeNET_{MS} 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. Statt dessen werden verschiedene Replikationen eines Simulators gestartet, die alle das vollständige Modell simulieren. Eine Synchronisation zwischen den Replikationen ist dann nicht notwendig. Ein zentraler 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.

Mit Hilfe der Simulation wurde das Modellfertigungssystem nun untersucht. Dabei stand die erreichbare Produktionsmenge im Vordergrund. Eine erste Untersuchung mit einfacher Transportstrategie des Regalförderers und nur einer Palette zeigte über eine Untersuchung der Aufenthaltswahrscheinlichkeiten der Werkstücke, daß das Entfernen fertiggestellter Produkte aus der rechten Übergabestation der größte Engpaß des Systems war. Offensichtlich mußte dem Transport leerer Paletten zu dieser Station eine höhere Priorität eingeräumt werden. Der Durchsatz betrug zunächst lediglich 10 Produkte pro Stunde. Jedes einzelne Werkstück hat eine gemessene Durchlaufzeit von ungefähr vier Minuten, so daß dieser Wert durchaus realistisch ist. Daraufhin wurde die Transportstrategie entsprechend verändert, so daß bei wartendem Werkstück auf dem Drehtisch sofort eine Palette zur Übergabestation befördert wird.

Die Simulation dieses Modells ergab keine signifikante Verbesserung. Im Anschluß wurde versucht, den Engpaß des Palettentransports durch eine Erhöhung der Palettenanzahl auf zwei und drei zu umgehen. Dadurch wurde der Durchsatz auf 12 bzw. 20 Werkstücke pro Stunde erhöht. Trotzdem ist das Hochregallager weiterhin der Engpaß des Systems. Es wurde deshalb mit einem Simulationsexperiment berechnet, wie sich der Durchsatz bei einer fiktiven Änderung der Geschwindigkeit des Regalförderers entwickeln würde. In einem realen Fertigungssystem könnte dann anhand der sich dadurch ändernden Investitionskosten und Mehreinnahmen eine optimale Auslegung bestimmt werden.

Das Werkzeug TimeNET_{MS} erlaubt die automatische Durchführung einer Reihe von Leistungs-

bewertungen. Dazu kann ein Parameter des Modells (im Beispiel ein Geschwindigkeitsfaktor für den Regalförderer) linear oder logarithmisch einen angegebenen Wertebereich durchlaufen. Für jeden Einzelwert wird eine Leistungsbewertung durchgeführt und die Ergebnisse im Anschluß mit *gnuplot* als Kurve dargestellt. Abb. 6 zeigt das Ergebnis der Berechnungen. Es wurde angenommen, daß durch technische Maßnahmen die Geschwindigkeit des Regalförderers um den Faktor zwei beschleunigt bzw. bis zum Faktor fünf verlangsamt werden könnte. Den jeweils resultierenden Durchsatz zeigt die Kurve. Dabei fällt auf, daß sich bei einer Verringerung der Transportdauer unter 60 Prozent des realen Wertes der Durchsatz nicht mehr verbessert. Ab diesem Punkt ist das Hochregallager offensichtlich nicht mehr der beschränkende Engpaß.

Die Simulationsläufe wurden mit einem Konfidenzniveau von 95 Prozent und einem maximalen relativen Fehler von 10 Prozent durchgeführt. Bis zum Erreichen dieser Genauigkeit wurden jeweils ungefähr drei Tage der Modellzeit durchlaufen, was auf einem Cluster von zehn Workstations 25 Sekunden dauerte.

5 Steuerung

Um mit Hilfe eines Petri-Netzes eine Steuerung realer Prozesse durchzuführen, müssen zu dem ansonsten autonomen Modell Möglichkeiten der Kommunikation mit der „Außenwelt“ vorgesehen werden. Dazu sind aus Modellsicht Eingangs- und Ausgangssignale erforderlich. Diese sollten möglichst einfach und ohne Bruch mit dem Verständnis des autonomen Modells hinzugefügt werden. Da die hier verwendeten farbigen Modelle streng an der tatsächlichen Struktur des Fertigungssystems orientiert sind, ist dies problemlos möglich.

Dabei wird folgende Auffassung vertreten: steuerbar sind nur aktive Betriebsmittel wie Maschinen, Transportmittel etc. Deren Aktivitäten werden durch Transitionen modelliert, die Marken entweder bewegen (Transport) oder ihre Farbe ändern (Bearbeitung). Eine Transition wird schaltfähig, sobald ihre Schaltbedingung erfüllt ist, die zu verbrauchenden Marken vorhanden sind und nachfolgende Stellen genügend freie Kapazität aufweisen. Von diesem Zeitpunkt an beginnt die Schaltzeit der Transition im zeiterweiterten Modell zu laufen. Erst nach Ablauf der Schaltzeit werden die Marken durch das Schalten bewegt. Nun wird einfach jedem einzeln ansteuerbaren Vorgang eine Transition zugeordnet. Wird die Transition schaltfähig, geht ein entsprechendes Startsignal an den Vorgang (z.B. ein Motor wird eingeschaltet). Ist die Aktivität abgeschlossen (z.B. ein Sensor zeigt die Endlage an), wird ein Signal an das Modell gesendet, welches die Transition zum sofortigen Schalten zwingt. Damit wird während einer laufenden Steuerung lediglich die im Modell unbekannte Dauer des Vorgangs übernommen.

Transitionen mit zugeordneten Eingangssignalen werden *extern* genannt, alle anderen *intern*. Die Zuordnung von Ausgangssignalen zu einer beliebigen Transition ändert ihre Schaltsemantik im Modell nicht. Externe Transitionen dagegen schalten genau dann, wenn ihr Eingangssignal empfangen wurde und sie im Modell normal schaltfähig sind. Die im Modell spezifizierte Schaltzeit einer externen Transition spielt beim laufenden Betrieb daher keine Rolle. Transitionen dieses Typs sollten nicht durch das Schalten anderer Transitionen deaktiviert werden können, denn dann würde ihr später ankommendes Eingangssignal nicht mehr bearbeitet wer-

den können. Konflikte dieser Art können anhand der Modellstruktur erkannt und als Warnung angezeigt werden.

Es ist möglich, einer Transition beliebig viele Eingangs- und Ausgangssignale zuzuordnen. Alle Ausgangssignale werden dann gleichzeitig beim Beginn der Schaltfähigkeit ausgesendet, während die Ankunft irgendeines Eingangssignals die Transition schaltet. Als Sonderfall sind auch Transitionen mit Ausgangssignal aber ohne Eingangssignal (oder umgekehrt) zugelassen. Ein Vorgang, der ohne explizites Startsignal jederzeit abgeschlossen sein kann, ist beispielsweise der Ausfall einer Maschine. Interne Transitionen behalten weiterhin ihre Schaltzeit, so daß z.B. Bearbeitungszeiten oder Zeitüberwachungen mit ihrer Hilfe realisiert werden können.

Diese Art der Anbindung an einen realen Prozeß unterscheidet sich von einem *steuerungstechnisch interpretierten Petri-Netz* [KQ88]. Bei diesen wird die Ausgabe über eine Abbildung der Markierung, also der Stelleninhalte, auf die Ausgangsgrößen vorgenommen. In der hier vorgeschlagenen Methode werden dagegen sowohl Ein- als auch Ausgaben über die Transitionen realisiert und als Signale aufgefaßt. Diese ähneln eher den Ereignissen (*events*) in *condition/event systems* [SK91]. Allerdings werden hier keine Bedingungen eingesetzt und es gibt Ereignissignale, die wie beschrieben an den Beginn und an das Ende der Schaltfähigkeit einer Transition gebunden sind.

Die Steuerung eines modellierten Fertigungssystems mit $\text{TimeNET}_{\text{MS}}$ kann auf zwei Arten vorgenommen werden, die beide direkt an die Funktion des Markenspiels gekoppelt sind. Wie in Abschnitt 3.4 beschrieben, kann damit einerseits eine interaktive Simulation vorgenommen werden, bei der der Benutzer angibt, welche der schaltfähigen Transitionen gefeuert werden soll. Alternativ kann das Markenspiel automatisch ohne Nutzereingriff ablaufen, wobei der Modellzustand und die feuernden Transitionen grafisch dargestellt werden. Bei beiden Möglichkeiten kann nun einfach die Steuerung des realen Systems zugeschaltet werden. Das Fertigungssystem wird dann entsprechend des Modellzustandes und der schaltfähigen Transitionen gesteuert. Der aktuelle Zustand ist im Modell sichtbar, und man kann jederzeit den automatischen Ablauf unterbrechen, um über die interaktive Simulation in das Geschehen einzugreifen. Damit ist die Steuerung transparent in das Werkzeug eingebunden, und das Modell dient gleichzeitig als Überwachungs- und Visualisierungsmittel. Ohne Bruch in der Beschreibungsmethode ist $\text{TimeNET}_{\text{MS}}$ damit auch als einfacher Leitstand zu verwenden.

Die beschriebene Funktionalität ist im Werkzeug $\text{TimeNET}_{\text{MS}}$ wie folgt realisiert: Während eines Markenspiels mit angeschalteter Steuerung überwacht ein zusätzlicher Prozeß den Modellzustand. Wird eine Transition neu aktiviert, kontrolliert der Prozeß, ob ihr Ausgangssignale zugeordnet sind. Falls ja, werden diese an die Hardware gesendet. Außerdem fragt der Prozeß ständig den Zustand der Sensoren im Fertigungssystem ab. Wird ein Zustandswechsel erkannt, der einem Eingangssignal zugeordnet ist, wird die entsprechende Transition geschaltet und der veränderte Modellzustand dargestellt. Ist vor der Ankunft eines solchen Signals die Schaltzeit einer internen Transition abgelaufen, wird diese geschaltet. Die Zuordnung von Transitionen zu Signalen ist in einer Datei beschrieben. Für die Steuerung durch ein Softwarewerkzeug wurde die in Abschnitt 2 beschriebene Modellfertigungsanlage durch elektronische Schaltungen zur Ansteuerung der Motoren, Abfrage der Sensoren sowie eine Kommunikationsschnittstelle erweitert. Diese sind aus Standardbausteinen modular und erweiterungsfähig aufgebaut. Details

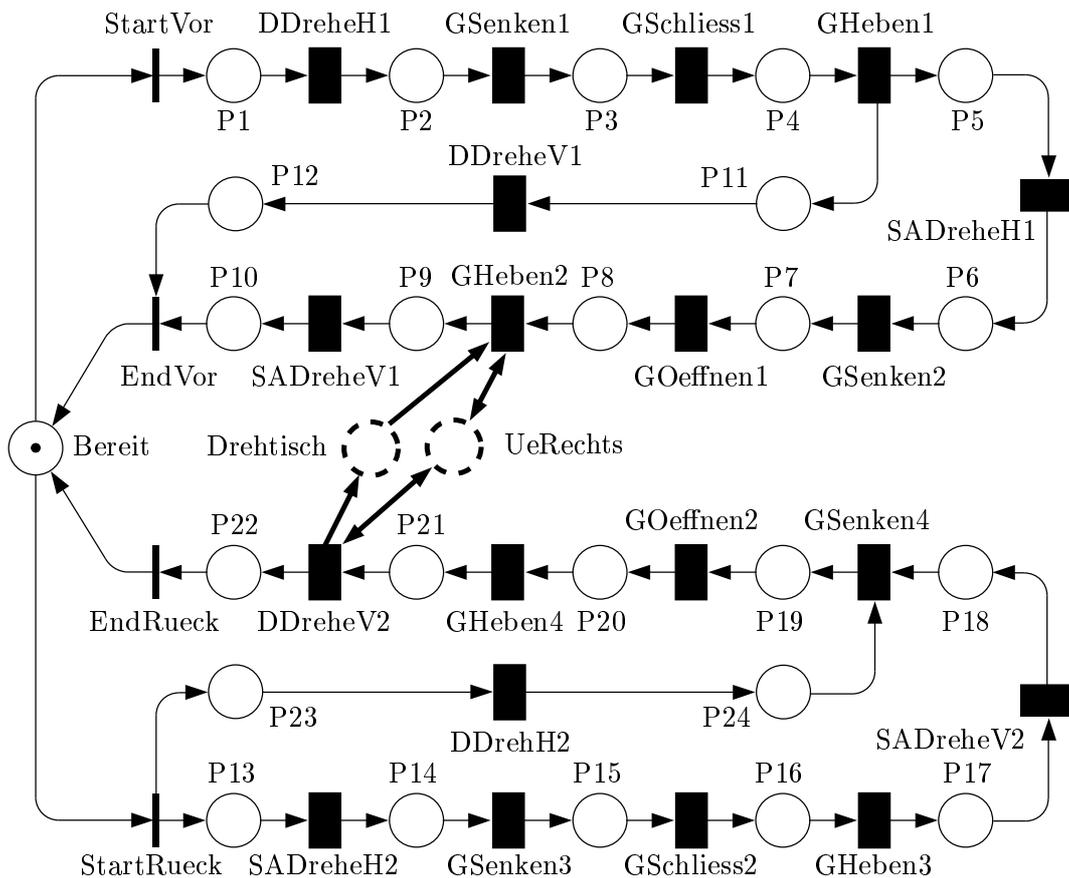


Abbildung 7: Verfeinertes Modell des Schwenkarmroboters.

zur Realisierung der Hard- und Software können [R98] entnommen werden.

Das in Abb. 7 gezeigte Petri-Netz ist als hierarchische Verfeinerung der Vertretertransition **Schwenkrob** aus Abb. 3 zugeordnet. Es modelliert das interne Verhalten des Schwenkarmroboters sowie die damit verbundene Ansteuerung des Drehtisches. Bis auf die Stellen **Drehtisch** und **UeRechts**, die Plätze für Werkstücke modellieren, werden ausschließlich Elementarstellen und -kanten benötigt. Diese sind wie im Bild sichtbar zur Unterscheidung dünn gezeichnet. Die genannten Produktstellen sind gestrichelt gezeichnet, da es sich hierbei um Anschlußstellen handelt, die nur Verweise auf die gleichnamigen Stellen auf der übergeordneten Modellseite darstellen.

Der Schwenkarmroboter kann zwei sinnvolle Aktionen durchführen: ein Werkstück vom Drehtisch zur rechten Übergabestation oder zurück transportieren. Der aktuelle Zustand des Roboters (und des Drehtisches) wird durch den Ort der ungefärbten Marke beschrieben. Im Bild ist der Ausgangszustand dargestellt, in dem sie sich in der Stelle **Bereit** befindet. Daran schließen sich zwei Transitionen **StartVor** und **StartRueck** an. Auf diese Art werden alle möglichen Startpunkte abgebildet. Durch die Verschmelzung mit den Informationen aus dem Arbeitsplan wird diesen Transitionen in ihren Übergangstabellen mitgegeben, für welche Werkstücke in den Stellen **Drehtisch** und **UeRechts** sie schalten dürfen. Wurden die Arbeitspläne fehlerfrei angegeben, kann der Schwenkarmroboter dadurch nur sinnvolle Transportaufträge beginnen. Fehler bei der Abbildung der Arbeitspläne oder falsche Transportstrategien lassen sich aber bei der

Untersuchung des Modells erkennen.

Mit **G** beginnende Transitionen beschreiben Aktionen des Greifers des Roboters (Heben, Senken, Öffnen und Schließen). Die Drehung des Schwenkarmes wird mit Transitionen modelliert, deren Namen mit **SA** beginnt, und Aktionen des Drehtisches beginnen mit **D**.

Nach dem Schalten einer der beiden Start-Transitionen beginnt die Abarbeitung genau eines Transportauftrags. Nacheinander werden deterministische Transitionen (gefüllte Rechtecke) schaltfähig und schalten. Sie beschreiben die einzelnen Schritte eines Transportablaufs, und ihnen sind Eingangs- und Ausgangssignale zugeordnet. Die Transition **DDreheH1** modelliert beispielsweise das Drehen des Drehtisches in die hintere Position. Dementsprechend ist ihr das Einschalten des Drehtisch-Motors in einer bestimmten Drehrichtung als Ausgangssignal und der entsprechende Endlagensensor (Mikrotaster) als Eingangssignal zugeordnet. Schaltet die Transition, ist der Drehtisch in der korrekten Position, um mit den folgenden Teilschritten fortzufahren. Transition **DDreheV1** zeigt, wie parallel ausführbare Operationen modelliert werden können. Die Transitionen **GHeben2** und **DDreheV2** bewegen bzw. ändern die Werkstückmarken in den Stellen **Drehtisch** und **UeRechts**. Damit wird modelliert, ab welchem Zeitpunkt die Werkstücke für andere Maschinen bereitstehen, auch wenn die Ausgangslage des Schwenkarmroboters erst noch hergestellt werden muß.

Im Vordergrund der Entwicklung stand eine bis hin zur Steuerung durchgängige Methode. Dabei wird keine Steuerungssynthese vorgenommen, die aus dem erstellten Modell beispielsweise ein SPS-Programm erzeugen könnte, sondern eine direkte Steuerung durch das modellbasierte Werkzeug realisiert. Dadurch ist das Modell auch in der Betriebsphase des Fertigungssystems zur Visualisierung verwendbar.

Ausgehend von den hier verwendeten speziellen farbigen Petri-Netzen läßt sich die Aufgabe der Steuerung zweiteilen: Über die Angabe der Arbeitspläne zu einem Strukturmodell wird zunächst erreicht, daß das Fertigungssystem (im Modell) die benötigten Produkte herstellt. Das bei der automatischen Übersetzung beider Modellteile entstehende Gesamtmodell muß aber weder verklemmungsfrei sein noch optimal bezogen auf bestimmte Leistungskenngrößen wie den Durchsatz. An dieser Stelle können anhand des Modells und der Methoden zur Untersuchung und Leistungsbewertung bessere Steuerstrategien gefunden werden. Diese können in einem beliebigen Zustand des Fertigungssystems lediglich die möglichen Operationen einschränken. Im Modell entspricht das einer Deaktivierung von schaltfähigen Transitionen. Dies ist in den verwendeten Petri-Netzen über die Angabe von markierungsabhängigen Schaltprädikaten (*guards*) sehr einfach möglich. Die zweite Stufe des Steuerungsentwurfs korrigiert und optimiert somit den Fertigungsablauf. Die anschließende direkte Steuerung des Fertigungsprozesses setzt das Modellverhalten in die Realität um.

Das in der beschriebenen Methode zunächst erstellte Modell der Struktur des Fertigungssystems muß alle strukturell möglichen Transport- und Bearbeitungsoperationen enthalten, um den Arbeitsplänen keine künstlichen Beschränkungen aufzuerlegen. Es ist daher mit einem Modell des ungesteuerten Prozesses vergleichbar, das für eine Verifikation der Steuerung notwendig ist [H97]. Allerdings wurde in den vorgestellten Modellen darauf verzichtet, tatsächlich *alle* (und damit auch sinnlose) Zustandsänderungen und Aktivitäten der Betriebsmittel zu modellieren.

Unter der Voraussetzung, daß kein Betriebsmittel ohne Steuersignal zu arbeiten beginnt, ist dies ausreichend. Können solche Fälle auftreten und sind sie für den modellierten Fertigungsprozeß von Belang, müssen sie natürlich mit abgebildet werden. Entscheidungen dieser Art machen ja gerade den Prozeß der Modellierung aus. Völlig auszuschließen wären Verklemmungen nur dann, wenn sich formal zeigen ließe, daß das Fertigungssystem korrekt und vollständig im Modell abgebildet ist. Dies ist jedoch leider nicht möglich.

6 Zusammenfassung

In diesem Beitrag wurde das Software-Werkzeug TimeNET_{MS} vorgestellt, mit dem Fertigungssysteme modelliert, untersucht und gesteuert werden können. Dazu wird in allen Phasen des Entwurfs und Betriebs eine spezielle Art farbiger zeiterweiterter Petri-Netze verwendet. Mit diesen ist die komfortable Modellierung der Struktur und Arbeitspläne getrennt möglich. Mit Hilfe der Untersuchung des Modells kann das geplante Fertigungssystem beispielsweise auf Verklemmungen untersucht werden. Außerdem ist eine Bewertung der Leistungsfähigkeit unter Berücksichtigung von Ausfällen und der gewählten Steuerstrategien möglich. Nach einer Optimierung durch den Vergleich verschiedener Varianten kann das Modell direkt zur Steuerung verwendet werden, indem an die Transitionen Eingangs- und Ausgangssignale gekoppelt werden.

Das vorgestellte Softwarewerkzeug ist unter den Betriebssystemen SunOS 5 (Solaris 2) lauffähig. Die neu implementierten Komponenten wurden in C++ realisiert, und die Benutzungsoberfläche verwendet das *Motif-Toolkit*. Es wird als Teil der demnächst erscheinenden Version 3 von TimeNET kostenfrei für nichtkommerzielle Zwecke angeboten. Weitergehende Informationen über TimeNET können unter der URL <http://pdv.cs.tu-berlin.de/forschung/timenet.html> abgerufen werden. Die Autoren bedanken sich bei den Studenten Frank Bayer, Mario Beck, Stefan Bode, Knut Dalkowski, Alexander Huck, Kolja Koischwitz, Andreas Kühnel, Stefan Rönisch und Ken Schwiersch für die Mitarbeit an der Realisierung des Softwarewerkzeugs.

Literatur

- [CGL94] Ciardo, G., German, R., Lindemann, C.: *A Characterization of the Stochastic Process Underlying a Stochastic Petri Net*. IEEE Transactions on Software Engineering **20** S. 506–515, 1994.
- [G94] German, R.: *Analysis of Stochastic Petri Nets with Non-Exponentially Distributed Firing Times*, Dissertation, Technische Universität Berlin, 1994.
- [GKZH95] German, R., Kelling, C., Zimmermann, A., Hommel, G.: *TimeNET – A Toolkit for Evaluating Non-Markovian Stochastic Petri Nets*. Performance Evaluation **24** S. 69–87, 1995.
- [H97] Hanisch, H.-M.: *Netzbasierte Methoden zum Steuerungsentwurf*. 5. Fachtagung Entwurf komplexer Automatisierungssysteme, S. 19–40, Braunschweig, 1997.

- [HFZ98] Huck, A., Freiheit, J., Zimmermann, A.: *Convex Geometry Applied to Petri Nets: State Space Size Estimation and Calculation of Traps, Siphons, and Invariants*. Zur Veröffentlichung eingereicht, 1998.
- [J92] Jensen, K.: *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1992.
- [K95] Kelling, C.: *Simulationsverfahren für zeiterweiterte Petri-Netze*, Dissertation, Technische Universität Berlin, 1995, Advances in Simulation, SCS International.
- [KQ88] König, R., Quäck, L.: *Petri-Netze in der Steuerungs- und Digitaltechnik*, Verlag Technik, 1988.
- [M89] Murata, T.: *Petri Nets: Properties, Analysis and Applications*. Proceedings of the IEEE **77** (4), S. 541–580, 1989.
- [R98] Rönisch, S.: *Steuerung einer Modell-Fertigungsanlage mit speziellen farbigen Petri-Netzen*. Diplomarbeit, Technische Universität Berlin, 1998.
- [ST96] Silva, M., Teruel, E.: *Petri Nets for the Design and Operation of Manufacturing Systems*. Proc. 5th Int. Conf. on Computer Integrated Manufacturing and Automation Technology (CIMAT '96), Lille, 1996.
- [SK91] Sreenivas, R. S., Krogh, B. H.: *On Condition/Event-Systems with Discrete State Realisations*. Proc. Discrete Event Dynamic Systems 1, S. 209–236, 1991.
- [ZDH96] Zimmermann, A., Dalkowski, K., Hommel, G.: *A Case Study in Modeling and Performance Evaluation of Manufacturing Systems Using Colored Petri Nets*. Proc. of the 8th European Simulation Symposium, S. 282–286, Genoa, 1996.
- [ZKH98] Zimmermann, A., Kühnel, A., Hommel, G.: *A Modelling and Analysis Method for Manufacturing Systems Based on Petri Nets*. Computational Engineering in Systems Applications (CESA '98), Nabeul-Hammamet, S. 276–281, 1998.
- [Z98] Zimmermann, A.: *Modellierung und Bewertung von Fertigungssystemen mit Petri-Netzen*, Dissertation, Technische Universität Berlin, 1997.