Bernd Däne, Angela Mölders, Andrea Melber, Wolfgang Fengler

Modellierung industrieller Prozesse mit Gefärbten Petri-Netzen

1 Einleitung

Dieser Beitrag beschreibt Erfahrungen, die bei der Erstellung und Realisierung von Modellen industrieller Transporteinrichtungen gewonnen wurden. Die Benutzung Gefärbter Petri-Netze [Jensen92] führt zu kompakten Modellen komplexer Prozesse. Um nützliche Auswertungen vornehmen zu können, enthält das Modell Stimulations- und Beobachtungskomponenten, welche die Prozeßumgebung repräsentieren. Die Netze verwenden Zeitattribute und Schaltwahrscheinlichkeiten. Die Auswertung erfolgt durch formale Analysen und durch Simulation. Die Implementierung von Steuerungssoftware erfordert die Partitionierung des Netzmodells. All diese Schritte werden beispielhaft am Modell eines sogenannten "Twin-Speed-Förderers" demonstriert.

2 Beispiel: Der Twin-Speed-Förderer

Der Twin-Speed-Förderer ist ein Förderband für den Transport von Stückgut, wie z.B. Kartons oder Kisten. Er besteht, wie in Abb. 1 gezeigt, aus drei parallelen Bändern. Die äußeren Bänder laufen mit geringerer, das innere mit höherer Geschwindigkeit. Da das innere Band etwas tiefer als die äußeren liegt, bewegen sich die Kartons normalerweise mit der geringeren Geschwindigkeit. Das innere Band kann jedoch durch einzeln ansteuerbare Stößel angehoben werden. Ein Karton, der sich auf der Position eines gerade aktivierten Stößels befindet, kann dadurch beschleunigt werden.

Der Twin-Speed-Förderer kann genutzt werden, um einen Strom zufällig eintreffender Kartons in einen kontinuierlichen Strom umzuwandeln, ohne daß sich die Kartons direkt berühren. Es soll also ein äquidistanter Teilestrom erzeugt werden, dessen Geschwindigkeit variieren kann. Diese Eigenschaft wird z.B. für die Adaption bestimmter Verpackungsmaschinen benötigt.

Um ein solches Verhalten zu erreichen, müssen die Stößel gezielt so angesteuert werden, daß sich die Abstände zwischen den Kartons auf dem Fließband verringern. Weiterhin muß die Geschwindigkeit der äußeren Bänder so gesteuert werden, daß "Über- und Unterläufe" des Gerätes vermieden werden.

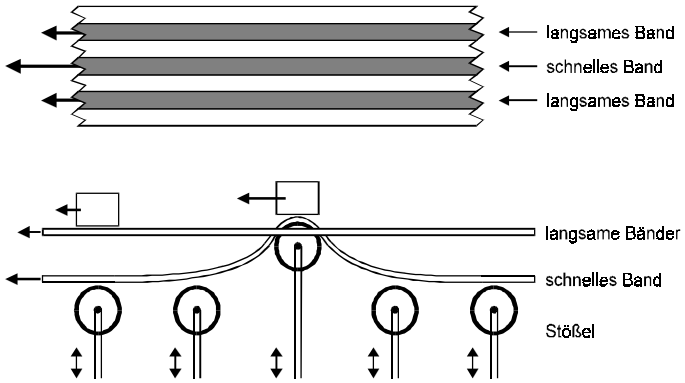

Abb. 1: Anordnung der Transportbänder

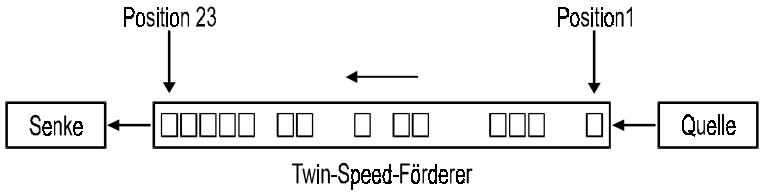

Abb. 2: Numerierung der Positionen

3 Modellierung des Beispielproblems mit Gefärbten Petri-Netzen

Zunächst wird eine Abstraktion des Problems benötigt, so abstrakt wie möglich und so konkret wie nötig. Dazu wird ein diskreter Zeitschritt ("Takt") eingeführt. Die maximale Ausgaberate des Gerätes soll "ein Karton pro zwei Takte" betragen. Der Twin-Speed-Förderer wird in 23 Positionen eingeteilt. Die Länge jeder Position entspricht der Summe aus der Länge eines Kartons und dem minimalen Abstand. Die Positionen sind entsprechend Abb. 2 numeriert.

Jede Position außer der letzten ist mit einem Stößel ausgestattet, der den Karton bei Aktivierung veranlaßt, sich auf die nächste Position zu bewegen. Dieser Vorgang erfordert einen Takt. Damit ist die Geschwindigkeit des inneren Bandes eine Position pro Takt. Das äußere Band kann 2, 4, 6, 9 oder 14 Takte benötigen, um sich um eine Position vorwärts zu bewegen, und es kann auch anhalten.

Im Petri-Netz-Modell wird jede Position des Bandes durch einen Platz repräsentiert ($p3_i$ in Abb. 3), der bei Vorhandensein eines Kartons auf dieser Position markiert wird. Zwei Gruppen von Transitionen modellieren die Bewegungsmöglichkeiten der Kartons. Eine dieser Gruppen ($t4_i$) wird simultan aktiviert und stellt die normale Bewegung dar, die andere ($t6_i$) repräsentiert die individuell aktivierbaren Stößel. Damit entsteht eine reguläre Struktur aus 13 Plätzen und 25 Transitionen, die auszugsweise in Abb. 3 dargestellt ist. Diese wird durch Färben in eine kompakte Form gefaltet, welche die Elemente p3, t4, t5 und t6 in Abb. 4 umfaßt. Die Markenfarben werden korrespondierend zu den Positionen mit <1> bis <23> bezeichnet.

Dieser Modellkern wird zu einem kompletten Prozeßmodell erweitert, indem Aspekte der Prozeßumgebung hinzugefügt werden. Hier wird nur das abstrakte Verhalten modelliert, nicht jedoch die Struktur. Als Senke für die transportierten Kartons dient p9. Als stochastische Quelle wurde eine Struktur mit p1, t1 und t2 implementiert. Es werden in zufälligen Abständen Marken erzeugt, deren durchschnittliche Rate ebenfalls zufällig zwischen zwei Werten wechselt. Die Elemente p2 und t3 passen das Förderband an die Quelle an.

Der nächste Schritt ist die Generierung einer Prozeßsteuerung. Diese hat die Kartons auf dem Förderband zu beobachten, die Stößel anzusteuern und die Geschwindigkeit der äußeren Bänder zu regulieren. Im Beispiel reicht

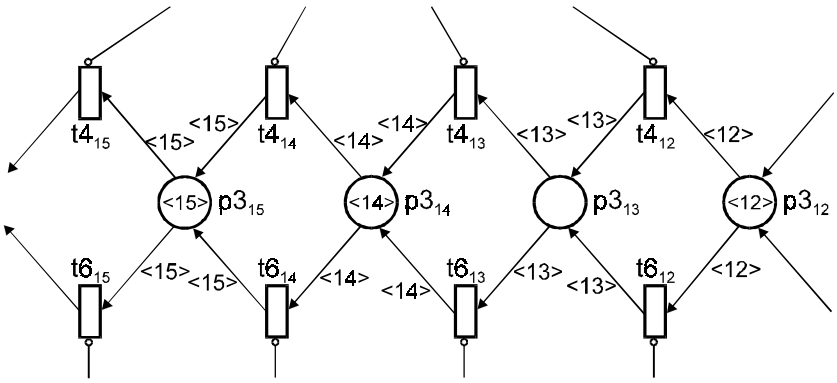

Abb. 3: Ausführliche Netzstruktur (Ausschnitt)

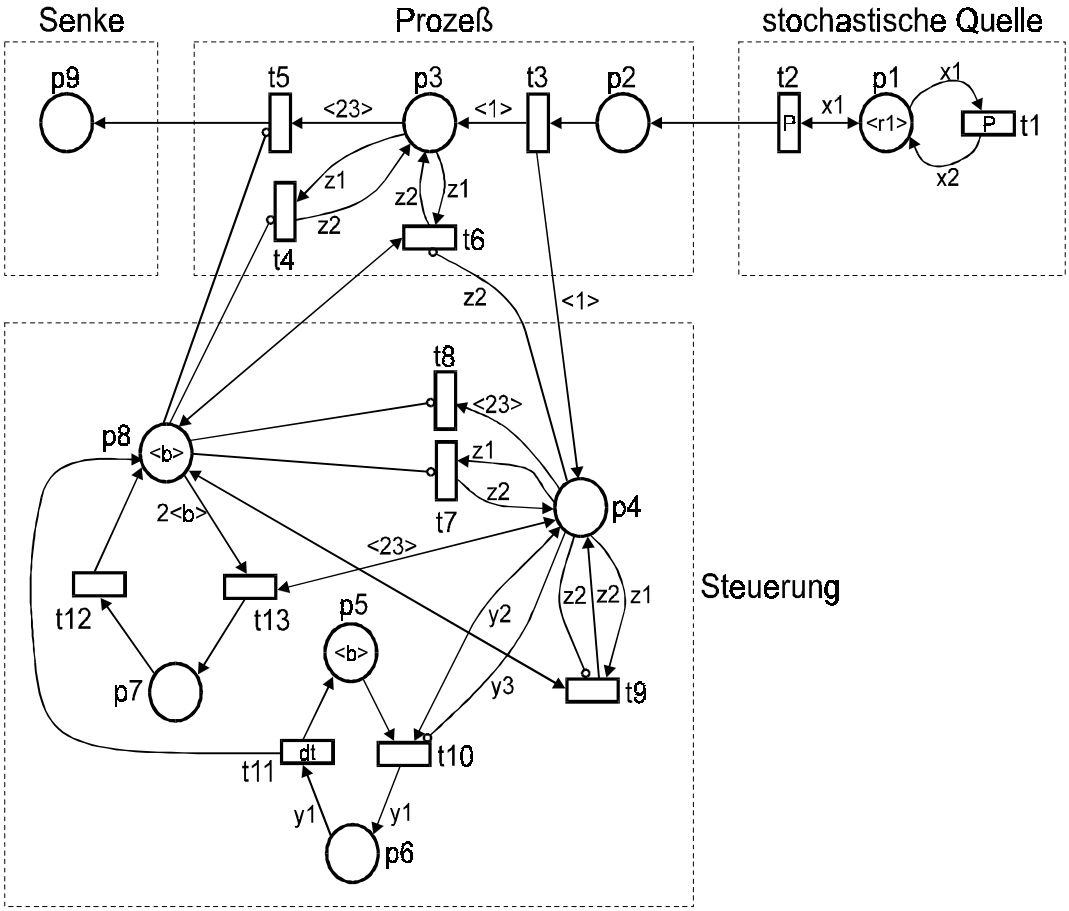

Abb. 4: Vollständiges Netz (siehe auch die Tabellen am Ende)

für die Beobachtung ein einziger Sensor aus, welcher die Ankunft eines Kartons auf der ersten Position meldet. Die volle Information über den Prozeßzustand wird im Inneren der Steuerungskomponente aus einem dort existierenden Abbild des Prozesses hergeleitet. Aufgrund des uniformen Modellkonzepts braucht hierzu nur das Modell des Prozesses selbst kopiert zu werden. Es ergeben sich die Elemente p4, t7, t8 und t9.

Die Ansteuerung der Stößel wird aus dieser Prozeßkopie abgeleitet. Ein Stößel wird immer dann aktiviert, wenn sich ein Karton auf seiner Position befindet und die nächste Position leer ist. Die Geschwindigkeit der äußeren Bänder ergibt sich beim gewählten Algorithmus aus der Anzahl lückenlos belegter Positionen ab der Position 23. Transition t10 erzeugt eine Marke, welche die benötigte Geschwindigkeit entsprechend Tabelle 5 in der Farbe kodiert. Diese Information beeinflußt die Schaltzeit von t11. Der Zyklus aus p8, t13, p7 und t12 kontrolliert schließlich das Schalten der den äußeren Bändern zugeordneten Transitionen mit der entsprechenden Rate.

4 Auswertung des Modells

Formale Methoden ermöglichen die Ermittlung fundamentaler Eigenschaften des Netzes. Sie können durch Analysetools wie INA [Starke90] ausgeführt werden. So können beispielsweise Fehler, Inkonsistenzen und Regelverletzungen im Netz aufgespürt werden. Konflikte in der Steuerungskomponente enthüllen z.B. unvollständige Spezifikationen und zeigen die Notwendigkeit weiterer Entwurfsentscheidungen an. Weiterhin

können unerwünschte Deadlocks ermittelt und das Verhalten des Modells unter worst-case-Bedingungen untersucht werden.

Ein anderer Weg ist die simulative Untersuchung unter Benutzung von Petri-Netz-Simulatoren [Wikarski96]. Während der Simulation kann der Entwickler die Effekte von Änderungen am Modell unmittelbar beobachten. Dieser intuitive Zugang, der durch die wissenschaftliche Forschung oft unterschätzt wird, führt zu einem schnelleren Entwicklungsprozeß und macht es einfacher, Lösungen zu finden und Fehler zu vermeiden.

Die Simulation kann auch benutzt werden, um quantitative Eigenschaften des Prozesses vorherzusagen. Mit Hilfe statistischer Auswertetools können beispielsweise Durchschnittswerte für den erreichbaren Durchsatz oder für die Anzahl auf dem Förderband befindlicher Kartons gefunden werden.

5 Implementierung von Zielsoftware

Ein Ziel des Modellierungsprozesses kann die Schaffung einer Basis für die Implementierung von Steuerungssoftware sein. Dies erfordert eine klare Partitionierung des Modells in Prozeß, Prozeßsteuerung und Umgebung, wie in Abb. 4 bereits gezeigt. Das Innere der Steuerungskomponente kann durch Implementierungstools in eine Programmiersprache wie z.B. C/C++ [Unger92] oder APL/2 [Witter96] transformiert werden. Dabei wird entweder ein spezieller Petri-Netz-Interpreter erzeugt [Fengler91] oder die Struktur des Erreichbarkeitsgraphen in eine Programmstruktur transformiert.

Große Aufmerksamkeit erfordert das Interface zwischen der Steuerungskomponente und der Prozeßkomponente. Die an dieser Stelle zerschnittenen Kanten des Petri-Netzes müssen entsprechend ihrer Art und Richtung entweder in Eingabe- oder Ausgabeaktivitäten der Steuerungssoftware transformiert werden. Eine ausführlichere Darstellung hierzu ist in [Däne97] enthalten.

In diesem Schritt wird stärker als in den anderen eine manuelle Interaktion benötigt. Beispielsweise würde der am Beispiel dargestellte Algorithmus ein impulsbasiertes Ausgangsinterface für die Geschwindigkeitssteuerung der äußeren Bänder ergeben. Eine bessere Lösung wäre eine manuell ergänzte Ausgabeaktivität, welche einen Ausgangswert entsprechend der Markenfarbe in p6 erzeugt.

Die gewählte Modellierungsmethodologie enthält die Möglichkeit, Teile der Steuerungssoftware wahlweise auch in Hardware zu implementieren. Das kann sinnvoll sein für zeitkritische Teile (z.B. Feldbus-Ansteuerung [Nützel95]) oder für Komponenten mit hohen Zuverlässigkeitsansprüchen (z.B. “Not-Aus”-Funktionen) und führt zu einem Beitrag zur Methodologie des Hardware-Software-Co-Design.

6 Zusammenfassung und Ausblick

Diese Fallstudie zeigt, daß Petri-Netze in allen Entwurfsphasen industrieller Einrichtungen von Nutzen sein können. Ihr Hauptvorteil ist die Integration der sehr verschiedenen Aspekte und Ziele der unterschiedlichen Entwurfsschritte in ein homogenes Darstellungsmittel. Es ist klar, daß die menschlichen Aktivitäten nicht eliminiert werden können, aber der Entwerfer wird von mechanischer Arbeit entlastet und sein Blick auf eine systembezogene Betrachtungsweise gelenkt.

Zusätzliche Attributierungen, wie Fuzzy-Erweiterungen [Fengler96] oder markenbezogene Datentypen, erweitern die Modellierungsmächtigkeit, behindern jedoch die Auswertung durch formale Analysen. Sie müssen abhängig von der konkreten Anwendung sorgfältig erwogen werden. Eine andere Entwicklungsrichtung ist das Überdecken der Petri-Netz-Modelle durch ein applikationsspezifisches grafisches Front-End.

7 Tabellen

Farbname	Bedeutung	Plätze
<v1>, <v2>, … , <v5>	verschiedene Geschwindigkeiten der äußeren Bänder	p6
<r1>, <r2>	verschiedene Raten des Stimulus	p1
<1>, <2>, … , <23>	korrespondieren zu den Positionen auf dem Band	p3, p4
<b>	andere (‘black’)	p2, p5, p7, p8, p9

Tabelle 1: Farben

x1	x2	p(t1)	p(t2)
<r1>	<r2>	0,01	0,1
<r2>	<r1>	0,01	0,33

Tabelle 2: Farbtabelle für t1 und t2

z1	z2	
<i>	<i+1>	*i=[1,..,22]*

Tabelle 3: Farbtabelle für t4, t6, t7, t9

Kanten ohne Beschriftung: <b>
p: Schaltwahrscheinlichkeit (normal 1.0) **dt:** Schaltdauer (normal 1)

Tabelle 4: Attribute

y1	y2	y3	dt(t11)
<v1>	<23>	<21>/\<22>	13
<v2>	$\bigwedge_{i=21}^{23}$ <*i*>	<19>/\<20>	8
<v3>	$\bigwedge_{i=19}^{23}$ <*i*>	<17>/\<18>	5
<v4>	$\bigwedge_{i=17}^{23}$ <*i*>	$\bigwedge_{i=11}^{16}$ <*i*>	3
<v5>	$\bigwedge_{i=11}^{23}$ <*i*>	-	1

Tabelle 5: Farbtabelle für t10 und t11

Quellenhinweise

[Däne97] Däne, B.; Mölders, A.; Melber, A.;Fengler, W.:
Modeling an Industrial Transportation Facility with Coloured Petri Nets.
ICATPN'97: 2nd Workshop for Manufacturing and Petri Nets, Toulouse June 23-27, 1997

[Fengler91] Fengler, W.; Philippow, I.:
Entwurf industrieller Mikrocomputersysteme. Carl Hanser Verlag, München 1991

[Fengler96] Fengler, W.; Wendt, A.; Bogoljubow, J.; Däne, B.:
The Use of Fuzzy Coloured Petri Nets for Modeling and Simulation in Manufacturing.
ICATPN'96: 1st Workshop for Manufacturing and Petri Nets, Osaka June 24-28, 1996

[Jensen92] Jensen, K.: Coloured Petri Nets. Springer Verlag. Berlin, New York. 1992

[Nützel95] Nützel, J.; Fengler, W. :
Using Formal Technics for Fieldbus Protocol Implementation.
11th ISPE/IEE/IFAC Conference on CAD/CAM, Pereira Aug. 28-30, 1995

[Starke90] Starke, P. H.: Analyse von Petri-Netz-Modellen, Teubner Verlag Stuttgart. 1990

[Unger92] Unger, H.; Ben Achour, K.:
A Petri Net Based Method to the Design of Parallel Programs for a Multiprocessor System.
In: Lecture Notes of Computer Science 634. Springer Verlag Berlin, New York. 1992

[Wikarski96] Wikarski, D. (Ed.):
Petri Net Tools: A Comparative Study.
ISST-Bericht 39/96 der Fraunhofer-Gesellschaft e.G., Berlin, Dez. 1996

[Witter96] Witter, M.:
Übersetzungskomponente für ein Petri-Netz-Tool mit der Ziel-Programmiersprache APL/2.
Diplomarbeit 200-95D-054. TU Ilmenau 1996

Autorenangaben:

Dr.-Ing. Bernd Däne
Dipl.-Math. Angela Mölders
Dipl.-Wirtsch.-Inf. Andrea Melber
Prof. Dr.-Ing. habil. Wolfgang Fengler

Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Postfach 100565
D-98684 Ilmenau
Tel.: +49-3677-69-2825
Fax: +49-3677-69-1614
{bdaene;moelders;amelber;wfengler}@theoinf.tu-ilmenau.de
http://www.theoinf.tu-ilmenau.de/ra1/