# *COSTPN* zur Modellierung und Kontrolle rekonfigurierbarer Systeme

Martin Zaddach und Hermann de Meer
Fachbereich Informatik - Telekommunikation und Rechnernetze
Universität Hamburg

**Kurzfassung**

Heutige technische und ökonomische Systeme bieten meist eine Vielzahl von Eingriffsmöglichkeiten, um sie dynamisch beeinflussen zu können. Selbst einfache Modelle sind schwer analytisch auswertbar. Daher wird eine Erweiterung der *Stochastischen Petrinetze (SPN)* vorgestellt, welche Komponenten zur Beschreibung von dynamischen Rekonfigurationen zur Verfügung stellt. Zu diesem Zweck wird eine neue Transition eingeführt, welche die Möglichkeit bietet, zu einem kontrollierten Zeitpunkt von einem Zustand in einen anderen überzugehen. Diese Stochastischen Petrinetze werden *COntrolled STochastic Petri Nets (COSTPNs)* genannt. Ziel dieser COSTPNs ist es, in Abhängigkeit einer *Reward*funktion eine optimale Strategie zu finden und die Leistungskenngrößen in Abhängigkeit der angewandten Strategie für das untersuchte COSTPN zu ermitteln. Zur rechnergestützten Auswertung werden die COSTPNs in *Extended Markov Reward Models (EMRMs)* übersetzt. Dieses Vorgehen wird anhand des *Emergency Supply Model* demonstriert.

**Stichworte:** Stochastische Petrinetze, rekonfigurierbare Systeme, Dynamische Optimierung, optimale Strategie, Erweiterte Markovsche Entscheidungsmodelle, Emergency Supply Model

## 1 Einführung

Petrinetze sind ein geeignetes Werkzeug, um Steuerungen von Kontrollflüssen und Synchronistation von Prozessen und Systemen zu beschreiben. So lassen sich mit ihnen beispielsweise Nebenläufigkeiten in Kommunikationssystemen angemessen modellieren. Anfang der achtziger Jahre wurden diese Netze zu Stochastischen Petrinetzen (SPNs) durch Komponenten zur Beschreibung von Zustandsverweilzeiten ergänzt. Mit diesen Netzen lassen sich auch zeitliche Abläufe modellieren. Diese Stochastischen Petrinetze erweiterten Trivedi, Ciardo und Muppala zu Stochstischen Rewardnetzen (SRNs) und schließlich wurden für die Zustandsverweilzeiten auch nicht-Markovsche Verweilzeitverteilungen zugelassen.
Diese Modelle unterstützen jedoch keine Eingriffsmöglichkeit *von außen*. Bei großen technischen und ökonomische Systemen kann es erforderlich sein, daß die Randbedingungen des Systems dynamisch geändert werden müssen. Das hat zur unmittelbaren Konsequenz, daß nun verschiedene Eingriffsstrategien

bezüglich des erwarteten Gewinns beziehungsweise der erwarteten Kosten bewertet werden können. Dieses führt zur Suche nach der optimalen Strategie.
Aus diesem Grund wurden neue Modellklassen eingeführt. Auf der Ebene der Stochastischen Petrinetze sind die *COntrolled STochastic PetriNets* entstanden. Sie ermöglichen dynamische Entscheidungen zur Anpassung und Rekonfiguration von Systemen, ohne die anschauliche Darstellung der Petrinetze aufgeben zu müssen, und erlauben die algorithmische Berechnung wichtiger Performance- und Performability-Maße.
Um die COSTPNs numerisch auswerten zu können, wurde ein Algorithmus entwickelt, mit dem sich COSTPNs auf äquivalente *Extended Markov Reward Models* abbilden lassen [DüDM96]. EMRMs sind wenig anschaulich in der Darstellung und komplexer als die COSTPNs, stellen für die numerische Auswertung jedoch eine geeignete Repräsentation dar.
Für die rechnergestützte Auswertung von EMRMs steht ein geeignetes Softwaresystem (*PENELOPE*) zur Verfügung [SeDM97, SeDM96]. PENELOPE unterstützt die Auswertung und Berechnung der Performance-Maße, sowie die Ermittlung der optimalen Strategie. Diese Strategie kann sowohl für stationäre Maße, als auch für transiente Maße, d.h. Maße unter einer Startverteilung und einem endlichen Zeithorizont, berechnet werden.
Zunächst werden in Abschnitt 2 die *Extended Markov Reward Models* und in Abschnitt 3 die anschaulicheren *Controlled Stochastic Petri Nets* erklärt. In Abschnitt 4 wird die Abblidung von COSTPNs auf EMRMs auf beschrieben. Das *emergency supply model* wird als Beispiel für die vorangegangen Modelle in Abschnitt 5 vorgestellt. Dieses Lagerhaltungsmodell ist mit geringen Modifikationen anwendbar zur Kontrolle von Resourcen in Kommunikationssystemen. Für dieses Modell existiert bis heute keine analytische Lösung, jedoch durch die numerische Auswertung kann man die optimale Strategie ermitteln.

# 2 Extended Markov Reward Models

Um die für ein System relevanten Leistungskenngrößen modellieren zu können, benötigt man dafür angemessene Modelle. Eine mögliche Modellklasse bilden die *Extended Markov Reward Models (EMRMs)*, welche in [Maus90, DeMe92] untersucht und in dem Analysewerkzeug *PENELOPE* [SeDM97] implementiert worden sind. EMRMs sind wie folgt definiert [DüDM96]:
Sei $Z = \{Z(t), t \geq 0\}$ eine zeitkontinuierliche Markovkette mit endlichem Zustandsraum $\Omega$. Auf $\Omega$ wird eine *Reward*funktion $r(\omega)$ mit $r : \Omega \rightarrow \mathbf{R}$ bestimmt, so daß zum Zeitpunkt $t$ im Zustand $Z(t) \in \Omega$ der augenblickliche Ertrag *(Instantaneous Reward)* definiert wird durch $X(t) = r(Z(t))$. In dem endlichen Zeithorizont $[0..t)$ ist der *akkumulierte Reward* dann $Y(t) = \int_0^t X(\tau)d\tau$, in Abhängigkeit des gewählten Startzustands der Markovkette $Z$. Eine Markovkette $Z$ mit einer Rewardfunktion $r$ wird *Markov Reward Model (MRM)* genannt. Bei ergodischen MRMs konvergiert der augenblickliche Reward gegen den Reward im zeitlichen Mittel: $\lim_{t\rightarrow\infty} E[X(t)] = \lim_{t\rightarrow\infty} \frac{1}{t}E[Y(t)] =: E[X]$. Für die Leistungsanalyse von MRMs ist die Verteilungsfunktion $\Psi(y,t) := P(Y(t) \leq y)$ (*Performability*) von großer Bedeutung.
EMRMs bieten darüber hinaus auch die Möglichkeit der Optimierung rekonfigurierbarer Systeme. Rekonfigurationen sind Eingriffsmöglichkeiten, mit denen *von außen* ein Zustandswechsel des EMRM von einem Zustand in einen anderen

erzwungen wird.
EMRMs stellen dazu *Rekonfigurations*kanten zur Verfügung: Befindet sich das System in einem Zustand von dem eine Rekonfigurationskante ausgeht, so kann dieser Kantenübergang zur Rekonfiguration des Systems genutzt werden. Eine Strategie $S(t)$ definiert alle zu treffenden Entscheidungen zu einem Zeitpunkt $t$. Strategien können zeitabhängig oder zeitunabhängig sein. In letzterem Fall gilt $S = S(t)\,\forall t$. Eine Strategie $\hat{S}(t)$ ist optimal genau dann, wenn das Performability-Maß $\psi$ nicht kleiner ist, als unter irgendeiner anderen Strategie $S(t)$. Unter einer festen Strategie $S$ bezeichnet dann $X^S$ den Reward im zeitlichen Mittel, $Y_i^{S(t)}(t)$ den akkumulierten Reward zum Zeitpunkt $t$ bei Start in Zustand $i$, und $Y_i^S(\infty)$ den Reward bis zur Absorption in nicht ergodischen Modellen.
Darüber hinaus verfügen EMRMs auch über sogenannte *Vanishing States* (Sprungzustände), in denen die Verweilzeit gleich Null ist. Diesen Zuständen werden feste Rewards zugeteilt (*Pulse Rewards*).

# 3 Controlled Stochastic Petri Nets

Bei der Modellierung vieler technischer und ökonomischer Systeme können EMRMs schnell unübersichtlich werden. Dieses ist vor allen Dingen dann der Fall, wenn es sich um Lagerhaltungsmodelle oder Wartesysteme[1] handelt. Über der Anzahl der Warte- bzw. Lagerplätze vergrößert sich der Modellzustandsraum des EMRMs multiplikativ. Die numerische Auswertung und Optimierung von stochastischen Prozessen, die mit Hilfe von COSTPNs modelliert werden, hängt im wesentlichen von der Anzahl der Zustände des EMRMs ab. Daher sollte man bei der Modellierung beachten, daß die Verdopplung der Kapazität einer einzigen Stelle des COSTPNs, sogleich die Verdopplung der Zustände des EMRMs nach sich zieht. Diese Komplexität ist zugleich ein Vorteil der Darstellung der COSTPNs, die es ermöglicht, komplexe Systeme mit einer kompakten Darstellung zu repräsentieren.
Ähnlich wie die Markov Reward Modelle zu EMRMs durch Hinzufügen von Rekonfigurationskanten erweitert werden, so werden die Stochastischen Petrinetze durch einen neuen Transitionstyp erweitert, den Rekonfigurationstransitionen.
Ein COSTPN besteht aus einem 7-Tupel $(PN, T_1, T_2, T_3, W, Pr, Rew)$. Dabei ist $PN = (P, T, I, O, H, M_0)$ das zugrundeliegende Petrinetz. $P$ sind die Stellen, $T = T_1 \cup T_2 \cup T_3$ die paarweise disjunkten Transitionsmengen, $I \subset P \times T \times \mathbf{N}$ die gewichtete Menge der Eingangskanten, $O \subset T \times P \times \mathbf{N}$ die gewichtete Menge der Ausgangskanten, $H \subset P \times T \times \mathbf{N}$ die gewichtete Menge von Inhibitorkanten und $M_0$ die Startmarkierung. Inhibitorkanten sind Kanten, die, bei Vorhandensein von Marken entsprechend ihres Gewichts, das Schalten der Transitionen verbieten.
$T_1$ sind die Markovtransitionen, $T_2$ sind zeitlose Transitionen *(Immediate)* und $T_3$ die Rekonfigurationstransitionen. Zeitlose Transitionen sind Transitionen, welche bei Aktivierung ohne Zeitverzug schalten. $W : (T_1 \cup T_2) \to \mathbf{N}$ definiert die stochastische Komponente des COSTPNs und kann markierungsabhängig

[1] Regulierungssysteme zur Kontrolle von Dienstgüte (*Quality of Service*), wie z.B. der *Leaky Bucket*, lassen sich sehr anschaulich mit COSTPNs beschreiben.

sein. $W(t_i) = w_i$ beschreibt die Schaltrate (*firing rate*), falls $t_i$ eine Markovtransition ist, eine Gewichtung[2] der Transition im Falle einer zeitlosen Transition.
$Pr : T \to \mathbf{N}$ ist eine Prioritätsfunktion. Die Prioriät ist für Markovtransitionen und Rekonfigurationstransitionen 0, sie haben also niedrigste Priorität, zeitlose Transitionen haben eine Priorität von mindestens 1, schalten also bevorzugt. Aktivierungen von Transitionen hängen von diesen Prioritäten ab.
$Rew : RS \to \mathbf{R}$ definiert die Rewardstruktur. Dabei ist $RS$ die Menge der erreichbaren Markierungen des Netzes (Erreichbarkeitsmenge). $Rew$ wird durch die *Pulse Rewards* der zeitlosen Transitionen und durch die Rewardraten der Markovzustände festgelegt. Durch $Rew$ wird das Performability-Maß $\psi$ bestimmt.
Ein wichtiges Modellierungshilfsmittel ist die Prioritätsfunktion $Pr$. Da die zeitlosen Transitionen eine höhere Priorität als die der anderen Klassen haben, schalten sie immer vor diesen. Auf diese Weise lassen sich Rekonfigurationen verbieten.

# 4 Erzeugung von EMRMs aus COSTPNs

Für die numerische Auswertung von COSTPNs ist die Übersetzung des Modells in die Repräsentation der EMRMs entscheidend. Der von Düsterhöft und De Meer beschriebene Algorithmus [DüDM96] wird im folgenden kurz erklärt.
Er basiert auf dem Algorithmus von Trivedi, Ciardo und Muppala [TrCM91], der ein *Markov Reward Model (MRM)* aus einem Stochastischen Petrinetz erzeugt.
Der Basisalgorithmus sieht vor, zunächst den Erreichbarkeitsgraphen des SPNs zu generieren, wobei Markovkanten mit ihrer Schaltrate gewichtet werden, die zeitlosen Transitionen mit einer Wahrscheinlichkeit, die man erhält, wenn die Gewichte aller aktivierten zeitlosen Transitionen zu einem Wahrscheinlichkeitsmaß normiert werden. Die Rewardstruktur der Transitionen wird auf die Zustände abgebildet. Man erhält so den erweiterten Erreichbarkeitsgraphen. Darüber hinaus werden *Vanishing States* eliminiert und aus der Transitionsmatrix $U$ des MRMs wird eine infinitesimale Generatormatrix erzeugt.
Dieser Algorithmus muß für COSTPNs modifiziert werden. COSTPNs haben andere Aktivierungsregeln für Transitionen, Rekonfigurationstransitionen müssen in Rekonfigurationskanten übersetzt werden und *Pulse Rewards* sind bei *Vanishing States* gestattet, selbst bei transienter Analyse. Vanishing States müssen nicht eliminiert werden, nur *Vanishing Loops* sind in der derzeitigen Definition [Maus90] von EMRMs nicht erlaubt. Die Umwandlung der Transitionsmatrix $U$ in die infinitesimale Generatormatrix $Q$ muß verändert werden, da die EMRMs Pulse Rewards zulassen.
Der in [DüDM96] vorgestellte Algorithmus sieht das folgende modifizierte Verfahren vor. Zunächst wird der Erreichbarkeitsgraph des COSTPNs gemäß den Aktivierungsregeln gebildet. Der Erreichbarkeitsgraph wird in einen erweiterten Erreichbarkeitsgraphen umgewandelt, analog zum Basisalgorithmus. Die Rewardstruktur des COSTPNs kann nun auf den erweiterten Erreichbarkeitsgraphen abgebildet werden. Der letzte Schritt besteht aus der Elimination der Vanishing Loops. Der dafür eingesetzte Algorithmus identifiziert zunächst alle

[2] Anmerkung: Diese Gewichtung muß nicht notwendigerweise mit konkurrierenden zeitlosen Transitionen zu einem Wahrscheinlichkeitsmaß normiert sein.

Subgraphen mit Vanishing States, leitet Rekonfigurationskanten um, und wendet schließlich ein Standardverfahren zur Elimination von Loops an [TrCM91]. Gibt es in den Subgraphen Vanishing Loops mit Pulse Rewards, so werden diese Pulse Rewards neu berechnet und ein neuer Vanishing State mit diesen Pulse Rewards eingefügt. So läßt sich aus dem COSTPN ein EMRM gewinnen.

# 5 Eine Beispielstudie

Als Bespiel für die Modellierung mit COSTPNs und die Übersetzung dieser zu EMRMs soll das *emergency supply model* dienen, welches von Ton de Kok[3] wie folgt formuliert wurde:

*We consider a single stockpoint, which is replenished from a single supplier. The supplier has two supply modes: normal (mode 1) and emergency (mode 2). The lead time is exponentially distributed with mean $\mu_i^{-1}$, with $\mu_1 < \mu_2$. The demand for an item ist Poisson with rate $\lambda$, i.e. customers arrive according to a Poisson process and each customer asks for exactly one unit.*
*Each demand that cannot be satisfied from the stock ist backordered. Each backordered demand costs $p$ per time unit, each item on stock costs $h$ per time unit. We assume that $p > h$. Items ordered at the supplier cost $c_i$, with $c_1 < c_2$. We want to determine the optimal replenishment policy for this model.*
*...*
Dieses System eignet sich nicht nur zur Beschreibung von Lagerhaltungsproblemen, vielmehr können die Bestellungen auch als Zuweisungen von Resourcen in einem Kommunikationsystem angesehen werden. Das Lager repräsentiert in diesem Fall die Resourcen.
Für die Modellierung dieses Modells sind nun folgende Annahmen naheliegend. Das Lager (*Stock*) ist von endlicher Kapazität $N$. Ist das Lager leer und kommen Kunden an, deren Nachfrage nicht befriedigt werden kann (*Backordered Demands*), so werden deren Anfragen in einer Warteschlange festgehalten und nach $FCFS$ bedient. Die Annahme $FCFS$ ist aufgrund der Ununterscheidbarkeit der Kunden nicht zwingend. Die Warteschlange der Backordered Demands ist potentiell unendlich, für die Modellierung und anschließende numerische Auswertung werden sie aber auf $2N$ beschränkt. Diese Modellierungsannahme ist gerechtfertigt, da das System in diesen nun unberücksichtigten Zuständen weit von den günstigen Betriebszuständen entfernt ist, und Parameter und Strategien so gewählt werden sollten, daß das Eintreten von diesen ungünstigen Zuständen nur mit vernachlässigbar kleiner Wahrscheinlichkeit eintreten würde.
Da das Lager (*Stock*) ebenfalls eine Warteschlange darstellt, deren Zustandsraum zu dem der backordered demands disjunkt ist, d.h. eine der Warteschlangen immer leer ist, lassen sich beide zu einer Warteschlange $X_3$ der Größe $3N$ verschmelzen:

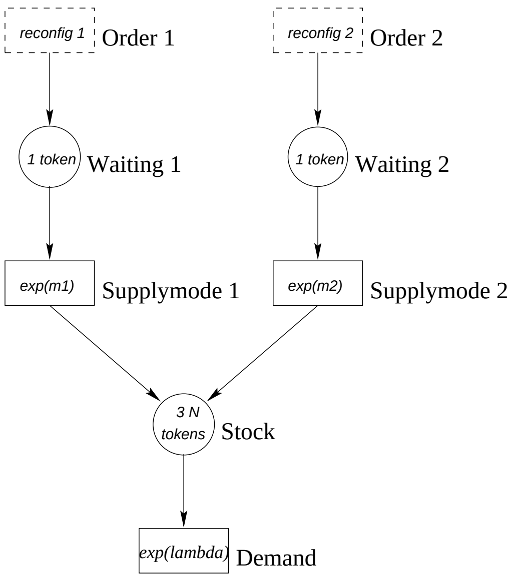
$$X_3 = x : \begin{cases} x - 2N & \text{Items im Lager} & \text{, falls } 2N \leq x \leq 3N \\ 2N - x & \text{Backordered Demands} & \text{, falls } 0 \leq x \leq 2N \end{cases}$$

[3]Kontakt: Dr. Ton G. de Kok, Professor of Operations Management, Department of Technology Management, Eindhoven University of Technology, P.O. Box 513 5600 MB Einhoven, the Netherlands

Da Nachbestellungen exponentiell und unabhängig voneinander verteilt sind, reicht für eine Markovsche Modellierung des Systems die Kenntnis aus, ob mindestens ein Item in jedem der Nachbestellungsmodi angefordert wurde. Für das COSTPN reicht also ein Zustandsraum $(X_1, X_2, X_3)$ aus, wobei $X_1$ und $X_2$ die beiden Supply Modes repräsentieren und $X_3$ die gemeinsame Warteschlange aus stock und backordered demands. Der zugrundeliegende Meßraum ist also $(E = \{0,1\}^2 \times \{0,\ldots,3N\}, \wp(E))$ mit der Größe $4(3N+1)$.
Nun muß noch die Rewardstruktur beschrieben werden. Für $X_3$ ergibt sich folgendes:

$$r_3(x) = \begin{cases} h(x-2N) & \text{, falls } 2N \leq x \leq 3N \\ p(2N-x) & \text{, falls } 0 \leq x \leq 2N \end{cases}$$

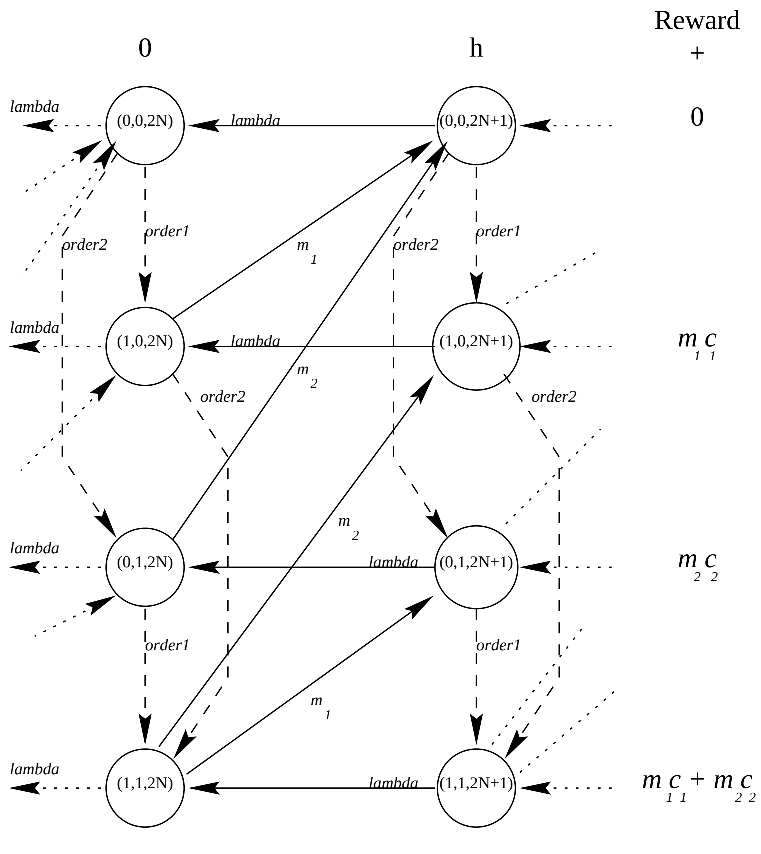
Für eine Warteschlangenlänge von $\{2N,\ldots,0\}$ ergeben sich Kosten von $\{0,\ldots,2Np\}$ pro Zeiteinheit gemäß der Anzahl der Backordered Demands, bei einer Länge von $\{2N,\ldots,3N\}$ sind die Kosten $\{0,\ldots,Nh\}$ pro Zeiteinheit gemäß der Anzahl der Items im Lager.
Hierbei ist zu beachten, daß Backorder Cost $p$ und Stocking Cost $h$ unter Umständen als Kosten negative Werte annehmen. Das ist vor allen Dingen dann der Fall, wenn das Analysetool nur Gewinnmaximierung unterstützt.
Die Bestellkosten $c_1$ und $c_2$, die bei Nachbestellungen zeitunabhängig anfallen, könnten als *Pulse Rewards* durch *Vanishing States* modelliert werden. Um das Modell aber möglichst klein zu halten, werden die Warteplätze der supply modes mit einer Rewardfunktion belegt: $r_i(x_i) = \mu_i \cdot x_i \cdot c_i$. Mit einer erwarteten Zwischenankunftszeit von $\mu_i^{-1}$ ergibt sich bei einer Nachbestellung $(x_i = 1)$ im Mittel ein Kostenquantum $c_i$. Das konstruierte COSTPN sieht nun wie folgt aus:




Figur 1: Emergency supply COSTPN

Hierbei sind die Transitionen *Order 1* und *Order 2* Rekonfigurationstran-

sitionen, die zu kontrollierten Zeitpunkten schalten können, während die *Supplymode 1*, *Supplymode 2* und *Demand* Markovtransitionen sind, deren Schaltzeiten $exp(\mu_1)$, $exp(\mu_2)$ und $exp(\lambda)$ verteilt sind. Die Stellen *Waiting 1* und *Waiting 2* haben eine Kapazitätsbeschränkung von 1 und einen Kostenreward von $r_1(\cdot)$ bzw. $r_2(\cdot)$. *Stock* hat eine Kapazität von $3N$ und einen Reward $r_3(\cdot)$. Der Gesamtreward ergibt sich für eine Markierung $m = (x_1, x_2, x_3)$ durch $r(m) = r_1(x_1) + r_2(x_2) + r_3(x_3)$.
Daraus läßt sich mit dem vorgestellten Verfahren das folgende EMRM gewinnen:




Figur 2: Emergency supply EMRM

Da in dem COSTPN keine zeitlosen Transitionen vorkommen, hat das EMRM nur Markovzustände. Jede Markierung des COSTPNs wird in einen Markovzustand des EMRMs übersetzt und dieses hat damit eine Größe von $4(3N+1)$. Die Namen der Zustände wurden aus Gründen der Übersichtlichkeit den Markierungen des COSTPNs angepaßt. Die Rekonfigurationstransitionen wurden in die Rekonfigurationskanten des EMRMs übersetzt. Die Anfangsmarkierung des COSTPNs bestimmt den Startzustand des EMRMs.
Nun kann man mit geeigneten Analysetools die optimale Strategie für das Modell ermitteln. Dabei erhält man für jeden Zustand bzw. für jede Markierung des Netzes die optimalen Rekonfigurationsstrategien. Diese können bei der

transienten Analyse zeitabhängig sein.
Softwaresysteme wie *PENELOPE* lassen darüber eine Variation der Parameter zu, so daß Rekonfigurationsentscheidungen in Abhängigkeit von den gewählten Parametern graphisch dargestellt werden können. Durch geeignete Variation von Parametern können auf diese Weise Erkenntnisse über das allgemeine unparametrisierte Modell gewonnen werden.

# 6 Zusammenfassung

Es wurden *Controlled Stochastic Petri Nets* als Modellierungskonzept für rekonfigurierbare Systeme auf der anschaulichen Abstraktionsebene der Petrinetze vorgestellt. Um die Performance- und Performability-Maße berechnen und eine optimale Strategie bestimmen zu können, werden diese COSTPNs auf *Extended Markov Reward Models* übersetzt, welche die gleiche Mächtigkeit, jedoch eine andere Abstraktionsebene besitzen. Sowohl stationäre als auch transiente Strategien können nun berechnet und die Kontrollentscheidungen mit Analysetools wie *PENELOPE* ausgewertet werden. Dieses Vorgehen wurde am Beispiel des *Emergency Supply Models* demonstriert. Für dieses Modell lassen sich nun optimale Strategien numerisch ermitteln, was analytisch bisher noch nicht möglich ist. Variationen von Parametern lassen Rückschlüsse auf das Modellverhalten zu, so daß man der analytischen Lösung näher kommt.

# Literatur


[DeMe92] de Meer, H.: "Transiente Leistungsbewertung und Optimierung rekonfigurierbarer fehlertoleranter Rechensysteme"; *Arbeitsberichte des IMMD der Universität Erlangen-Nürnberg* 25 (10), Oktober 1992

[DeMe94] de Meer, H., K. S. Trivedi, M. Dal Cin: "Guarded Repair von Dependable Systems"; *Theoretical Computer Science*, Special Issue on Dependable Parallel Computing 129, pp. 179-210, Juli 1994

[DüDM96] Düsterhöft, O.-R., H. de Meer: "Controlled Stochastic Petri Nets"; *16th Symposium on Reliable Distributed Systems*, Durham, North Carolina, pp. 18-25 (Oktober 1997)

[Maus90] Mauser, H. *Implementierung eines Optimierungsverfahrens für rekonfigurierbare Systeme* Studienarbeit 22/90 am IMMD IV, Universität Erlangen-Nürnberg, 1990

[SeDM96] Sevcikova, H., H. de Meer: "XPenelope User Guide for XPenelope Version 3.1"; FBI-HH-M-265/96, Universität Hamburg, Fachbereich Informatik, 1996

[SeDM97] Sevcikova, H., H. de Meer: "PENELOPE Dependability Evaluation and the Optimization of Performability"; *0th Intern. Conf. on Computer Performance Evaluation*, St. Malo, Frankreich, Lectures Notes on Computer Science (LNCS 1245), pp. 19-31 (Juni 1997)

[TrCM91] Trivedi, K.S., G. Ciardo, J. K.Muppala: "On the solution of GSPN reward models"; *Performance Evaluation 12*, pp. 237-253, 1991