

# Integration erweiterter Fehlerbäume und generalisierter stochastischer Petrinetze

Kerstin Buchacker

IMMD3, Friedrich-Alexander-Universität, Erlangen  
kerstin.buchacker@informatik.uni-erlangen.de

**Zusammenfassung** Sowohl Fehlerbäume als auch stochastische Petrinetze werden in der Zuverlässigkeitsanalyse zur Modellierung komplexer technischer Systeme eingesetzt. Fehlerbaummodelle sind klar und gut strukturiert, können jedoch wichtige Systemeigenschaften, wie Ausfall- und Reparaturabhängigkeiten zwischen einzelnen Bestandteilen des Systems ebensowenig erfassen wie Systemkomponenten mit mehr als zwei Zuständen. Es bietet sich daher an, diese Nachteile der Fehlerbäume durch die Kombination mit stochastischen Petrinetzen aufzuheben, welche sich gerade für die Modellierung komplexer stochastischer Abhängigkeiten gut eignen. Hier wird daher ein Verfahren vorgestellt, welches die Beschreibung eines technischen Systems mit Hilfe *erweiterter* Fehlerbäume mit der Auswertung des Modells auf der Grundlage stochastischer Petrinetze kombiniert.

## 1 Einleitung

Die Fehlerbaumanalyse (*Fault Tree Analysis, FTA*) ist ein Standardverfahren für Sicherheits- und Zuverlässigkeitsuntersuchungen, welches seit langem in der Industrie eingesetzt wird [6, 8]. Die klassische FTA gibt Antwort auf die Frage „Aufgrund welcher Ursachen ist ein (unerwünschtes/katastrophales) Ereignis eingetreten? Mit welcher Wahrscheinlichkeit?“ Die strukturierte graphische Darstellung des Modells als Fehlerbaum (*Fault Tree, FT*) ist auch für Außenstehende relativ einfach verständlich und nachvollziehbar. Mathematisch gesehen beschreiben FT das Zusammenwirken von Ereignissen bzw. die funktionalen Zusammenhänge einzelner Systemkomponenten als boolesche Funktion. Der FT wird ausgehend vom (unerwünschten) Top-Ereignis (TE) aufgebaut. Es werden rekursiv alle (Zwischen-)Ereignisse untersucht, deren Zusammenwirken das TE verursachen kann. Die Rekursion endet bei den Basisereignissen bzw. Einzelkomponenten, die die feinste Auflösung auf dem gewählten Abstraktionsniveau darstellen. Die FTA untersucht gezielt *alle* Möglichkeiten des Eintretens *eines* Ereignisses. Die qualitative FTA, welche Antwort auf die Frage nach den Ursachen des TE gibt, kann nur durchgeführt werden, falls sich alle Basisereignisse mit booleschen Variablen beschreiben lassen. Enthält ein System also Komponenten, deren Einfluß auf das TE sich nicht in den zwei Zuständen „intakt“ und „defekt“ ausdrücken läßt, so ist eine FTA nicht möglich. Damit die Wahrscheinlichkeit des TE berechnet werden kann (quantitative FTA) müssen zudem alle Basisereignisse paarweise stochastisch unabhängig sein. Gerade diese Forderung wird aber von den wenigsten realen Systemen erfüllt, so daß die Annahme der stochastischen Unabhängigkeit im Modell zu Ergebnissen führen kann, die stark von den tatsächlichen Werten abweichen.

Generalisierte stochastische Petri-Netze (GSPN,[1]) und *Stochastic Reward Nets (SRN, [3])* hingegen sind besonders geeignet zur Modellierung komplexer zeitbehafteter Zusammenhänge zwischen Ereignissen. In der Industrie beginnen GSPN in den Bereichen der Sicherheits- und Zuverlässigkeitsanalyse erst langsam Fuß zu fassen. Dies mag nicht zuletzt an den für Außenstehenden oft nur schwer überschaubaren GSPN liegen, die bei der Modellierung eines komplexen Systems entstehen können.

Es liegt daher nahe, FT und GSPN zu kombinieren, um die Vorteile beider Verfahren zu nutzen. Unterschiedliche Ansätze der Integration werden in [4, 7, 9] und [10] vorgestellt. In dem hier vorgeschlagenen Verfahren werden FT, die um einige Konstrukte zur einfachen Beschreibung komplexerer Basisereignisse sowie stochastischer Abhängigkeiten zwischen den Basisereignissen erweitert wurden, als Modellierungsschnittstelle verwendet, um die Vorteile der FT auf diesem Gebiet zu nutzen. Die Modellauswertung hingegen erfolgt — für den Anwender transparent — auf der Grundlage von GSPN, wodurch die Flexibilität von GSPN-Modellen zur Modellierung von Komponentenabhängigkeiten zur Verfügung steht.

Im folgenden Abschnitt wird zunächst die Modellierungsschnittstelle mit erweiterten FT anhand eines einfachen Beispiels (Abb. 1) dargestellt. Abschnitt 3 beschreibt die Auswertung der erstellten Modelle mit GSPN anhand des Beispiels und stellt Möglichkeiten vor, um die Größe des Zustandsraums der GSPN zu verringern.

## 2 Modellierungsschnittstelle

Die Modellierungsschnittstelle ermöglicht es dem Modellierer, den Aufbau des Systems sowie die Eigenschaften und das Zusammenwirken der einzelnen Bestandteile kompakt und übersichtlich zu beschreiben. Die Erstellung des komplexen GSPN-Modells geschieht automatisch anhand der Vorgaben des Modellierers. Das Modell ist dreigeteilt und besteht aus den Definitionen der Einzelereignisse bzw. -bestandteile, der Struktur des betrachteten Systems beschrieben durch den funktionalen Zusammenhang dieser Grundeinheiten als FT, sowie weiteren Angaben zu Abhängigkeiten zwischen den Einzelkomponenten, welche nicht in einem FT dargestellt werden können. Zur Modellierung steht momentan eine Textschnittstelle zur Verfügung, die die Beschreibung des Systems mittels einer kompakten formalen Sprache ermöglicht.

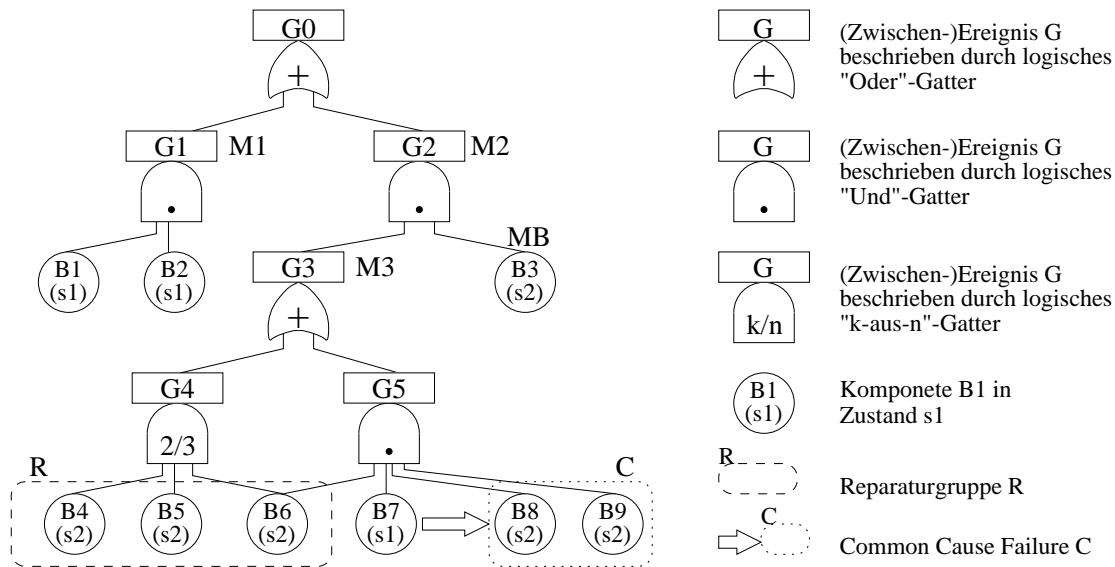


Abbildung1. Beispiel eines erweiterten Fehlerbaums

### 2.1 Modellierung der Basisereignisse

Da die Basisereignisse häufig Zustandswechsel einzelner Komponenten darstellen, werden hier die Begriffe „ausfallen“ bzw. „reparieren“ gleichgesetzt mit eintreten bzw. wieder verschwinden eines Basisereignisses. Die Modellierungsschnittstelle erlaubt Komponenten mit je einem intakten und mehreren Fehlzuständen. Letztere können sich entweder gegenseitig ausschließen oder aber verschiedene Stufen der Funktionseinschränkung (*degradation*) der Komponente darstellen. Schalter oder Ventile mit den Zuständen „intakt“, „klemmt offen“ und „klemmt geschlossen“ sind Beispiele für den ersten Fall. Pumpen oder Motoren hingegen können als degradierende Komponenten beispielsweise mit den Zuständen „intakt“, „1/2 Leistung“, „1/4 Leistung“ und „Stillstand“ modelliert werden.

Das Ausfallverhalten einer Komponente mit sich gegenseitig ausschließenden Fehlzuständen wird durch die Raten angegeben, mit denen sie vom Intakzustand aus in ihre verschiedenen Fehlzustände übergeht. Ein direkter Wechsel zwischen den Fehlzuständen ist bei dieser Art von Komponente nicht möglich. Degradierenden Komponenten wird für jeden Fehlzustand eine Rate zugeordnet, die angibt, wie schnell die Komponente weiter degradiert, d. h. aus dem momentanen Zustand in einen Zustand mit noch weiter eingeschränkter Funktionalität übergeht. Ist die Komponente reparierbar, so kann für jeden Fehlzustand die Reparaturrate angegeben werden, mit der die Komponente wieder in den Intakzustand übergeht. Für alle Komponenten gilt, daß sie sich zu jedem beliebigen Zeitpunkt in genau einem Zustand befinden müssen. Abbildung 2 zeigt die Definition der Komponenten B1 und B2 aus Abb. 1.

```

DEFINE COMPONENT B1 :
    STATES = s0 (INTACT),
             s1 (FRATE = 0.0003, RRATE = 0.2),
             s2 (FRATE = 0.0001, RRATE = 0.2);
END

DEFINE DEGRADING COMPONENT B2 :
    STATES = s0 (INTACT),
             s1 (FRATE = 0.00002, RRATE = 0.1),
             s2 (FRATE = 0.00002, RRATE = 0.1);
END
    
```

Abbildung2. Definition der Komponenten B1 und B2 aus Abb. 1

## 2.2 Modellierung der Systemstruktur

Dieser Teil des Modells beschreibt die Systemstruktur im wesentlichen als klassischen FT, der allerdings um die Möglichkeit erweitert wurde, Komponenten mit mehreren Zuständen zu erfassen. Die Systemstruktur wird durch boolesche Gatter modelliert, die das Zusammenwirken der Ereignisse beschreiben (Abb. 1). Es sind grundsätzlich alle booleschen Funktionen als Gatter möglich, meist beschränkt man sich jedoch auf eine Auswahl. In den hier betrachteten erweiterten FT sind „Und“- , „Oder“- , „Nicht“- und „ $k$ -aus- $n$ “-Gatter erlaubt. Die Blätter des Baumes repräsentieren in der klassischen FTA boolesche Variable, die den Wert 1 annehmen, wenn das Ereignis eingetreten (die Komponente ausgefallen) ist, und den Wert 0 sonst. Bei Multi-State-Komponenten beschreibt ein Blatt hingegen das Ereignis „Komponente B befindet sich in Zustand  $s$ “, außer dem Namen der Komponente muß daher der entsprechende Zustand im Blatt angegeben werden. Mathematisch muß eine Multi-State-Komponente durch eine mehrwertige Variable bzw. durch mehrere boolesche Variable dargestellt werden.

## 2.3 Modellierung stochastischer Abhängigkeiten

Stochastische Abhängigkeiten zwischen einzelnen Komponenten treten in realen Systemen häufig auf. Besonders gefürchtet sind *Common Cause Failures*, zeitgleiche Ausfälle mehrerer Einzelkomponenten aufgrund einer gemeinsamen Ursache. Aber auch die Mehrbelastung einer Komponente durch den Ausfall einer anderen kann deutliche Auswirkungen auf das Ausfallverhalten des Gesamtsystems haben. Bei reparierbaren Systemen sind außerdem Reparaturabhängigkeiten zu berücksichtigen, da im allgemeinen nicht davon ausgegangen werden kann, daß für jede Komponente eine eigene Reparaturmannschaft zur Verfügung steht.

Ausfallabhängigkeiten werden durch ihre Ursache und Wirkung charakterisiert. Die Ursache kann sowohl der Ausfall einer Komponente sein, als auch ein zusammengesetztes Ereignis, welches durch einen erweiterten FT beschrieben wird. Die Auswirkung ist der nachfolgende Übergang einer oder mehrerer Komponenten in einen bestimmten Fehlzustand. In Abb. 3 wird als Beispiel die *Common Cause Failure* aus Abb. 1 definiert, deren Ursache der Übergang der Komponente B7 in den Zustand  $s1$  ist.

Bei Reparaturabhängigkeiten muß zunächst jeweils die Anzahl der verfügbaren Reparaturressourcen und die Gruppe der Komponenten, die auf diese Ressourcen angewiesen sind, festgelegt werden. Außerdem müssen jeweils die Raten, mit denen der Ausfall einer Komponente erkannt und mit der Reparatur begonnen, sowie die Raten, mit der die Reparatur beendet wird, festgelegt werden. Als Beispiel wird in Abb. 3 die Reparaturgruppe R aus Abb. 1 definiert.

```
DEFINE COMMON CAUSE C :
    CAUSE   = B7 . s1;
    EFFECT  = B8 -> s2,
            B9 -> s2;
END

DEFINE REPAIR GROUP R :
    MEMBERS      = B4, B5, B6;
    RESOURCES     = 1;
    DETECTION RATE = GROUP : 0.2;
    COMPLETION RATE = GROUP : 0.4;
END
```

Abbildung3. Definition der *Common Cause Failure* (links) und der Reparaturgruppe (rechts) aus Abb. 1

## 3 Modellauswertung

Die quantitative Auswertung des Modells aus Abschnitt 2 und die Bestimmung der Wahrscheinlichkeit des TE geschieht in mehreren Schritten. Zunächst wird das Modell ganz oder teilweise (transparent für den Anwender) in GSPN transformiert. Diese werden mit dem am IMM3 zur Verfügung stehenden GSPN-Werkzeug PANDA [2] gelöst und die Ergebnisse anschließend zu einem Gesamtergebnis auf dem Abstraktionsniveau der Modellbeschreibung zusammengefaßt. Der folgende Abschnitt beschreibt anhand von Beispielen die Transformation der einzelnen Modellbestandteile in GSPN. Ein Nachteil der Verwendung von GSPN zur Modellauswertung ist ihr möglicherweise sehr großer Zustandsraum. Abschnitt 3.2 geht daher auf Verfahren zur Verkleinerung des Zustandsraumes der erzeugten GSPN ein.

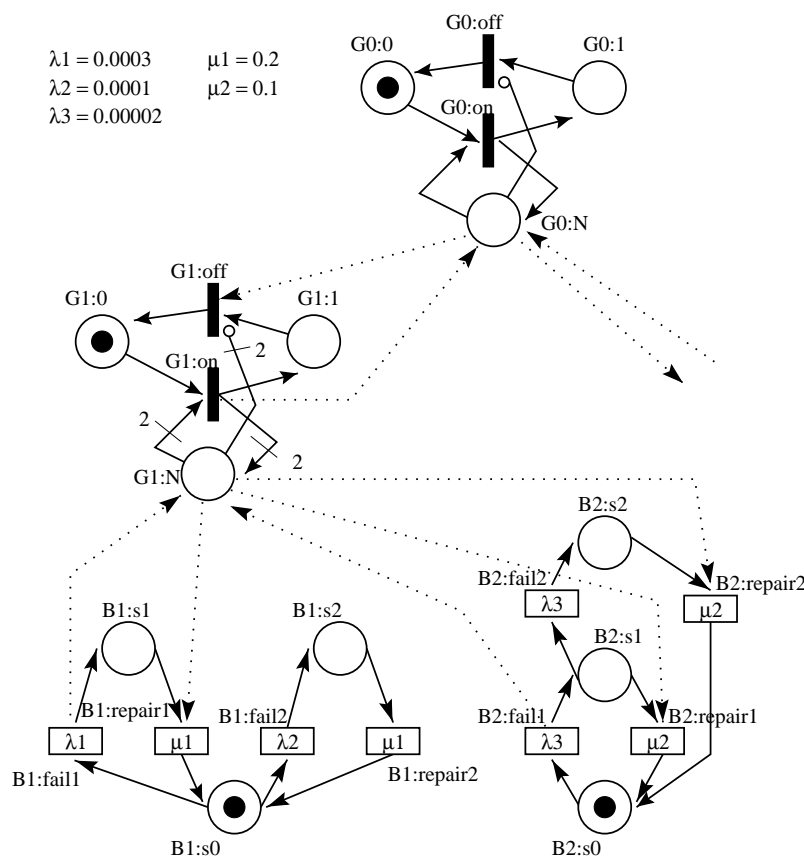
### 3.1 Transformation des Modells in GSPN

Die Transformation in GSPN geschieht automatisch und für den Anwender transparent. Die zur Lösung der erzeugten GSPN eingesetzten Werkzeuge müssen die Modellierung mit verbotenden Kanten und Kantengewichten erlauben. Als Anfangsmarkierung wird der Systemintaktzustand gewählt, in dem sich alle Einzelkomponenten jeweils in ihrem Intaktzustand befinden. Die Transformation wird in den folgenden Abschnitten anhand des Modells aus Abb. 1 anschaulich dargestellt.

**Transformation der Komponenten** Für jede Komponente wird anhand ihrer Definition ein einfaches GSPN generiert. Der untere Teil von Abb. 4 zeigt die Umwandlung der Komponenten B1 und B2 aus Abbildung 2 in ein GSPN. Für jeden Zustand einer Komponente wird eine Stelle im GSPN benötigt. Da

sich eine Komponente immer genau in einem Zustand befindet, enthält das Komponenten-Subnetz jeweils genau eine Marke. Die Übergänge zwischen den Zuständen werden durch zeitbehaftete Transitionen modelliert, welchen als Schaltraten die entsprechenden Ausfall- bzw. Reparaturraten der zugehörigen Komponente zugewiesen werden.

**Transformation des Fehlerbaums** Der FT kann ebenfalls vollständig in ein GSPN umgewandelt werden. Hierzu wird zu jedem Gatter ein entsprechendes einfaches GSPN erzeugt, welches je eine Stelle besitzt, um anzuzeigen, ob der Ausgang des Gatters 1 oder 0 ist, sowie eine zusätzliche Stelle, die mitzählt, wieviele Eingänge des Gatters 1 sind. Zwei zeitlose Transitionen verschieben in Abhängigkeit von der Eingangsbelegung des Gatters die Marke für die Ausgangsbelegung in die richtige Stelle. Alle Gatter-GSPN enthalten ausschließlich zeitlose Transitionen, da sich Zustandsänderungen des Systems immer ohne Zeitverzögerung im TE widerspiegeln müssen. Abbildung 4 zeigt einen Teilbaum aus dem Modell in Abb. 1 bestehend aus dem „Oder“-Gatter G0 und dem „Und“-Gatter G1 sowie den Komponenten B1 und B2 als GSPN (Anfangsmarkierung: alle Komponenten intakt). Die Gatter-GSPN und die Komponenten-GSPN werden analog der FT-Struktur miteinander verknüpft (gestrichelte Kanten in Abb. 4), so daß sich Zustandsänderungen in den Blättern bis zum TE fortpflanzen können. Verändert nun eine Komponente ihren Zustand durch Feuern der entsprechenden Transition, so feuert anschließend eine Folge zeitloser Transitionen. Ist keine zeitlose Transition mehr aktiviert, spiegelt die Markenbelegung im GSPN den Zustand der Gatter und Blätter im FT korrekt wider.



**Abbildung 4.** Transformation eines Teilbaums aus Abb. 1 in ein GSPN

**Transformation der stochastischen Abhängigkeiten** Die stochastischen Abhängigkeiten zwischen den Komponenten können direkt in das GSPN integriert werden. Dies wird am Beispiel zweier Komponenten aus der Reparaturgruppe R in Abb. 5 veranschaulicht, die sich eine gemeinsame Reparaturressource teilen. Eine Komponente kann nur repariert werden, wenn die Reparaturressource nicht gerade durch die andere Komponente belegt ist.

### 3.2 Verfahren zur Zustandsraumreduktion

Bei der Modellierung komplexer Systeme kann der Zustandsraum des erzeugten GSPN so groß werden, daß die Lösung des GSPN unverhältnismäßig aufwendig wird. Die folgenden Abschnitte stellen Verfahren vor, die die Generierung eines großen Zustandsraumes durch Ausnutzung der Fähigkeiten des GSPN/SRN-Werkzeugs PANDA vermeiden.

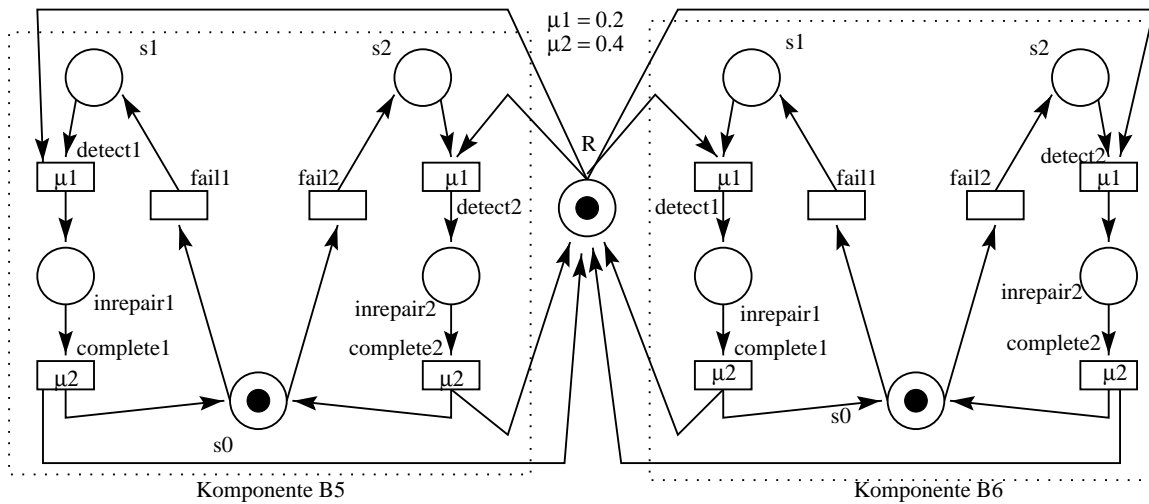


Abbildung5. GSPN der Reparaturgruppe R aus Abb. 3

**Reduktion der *vanishing* Markierungen** Bei dem oben beschriebenen Verfahren zur Erzeugung des GSPN werden insbesondere durch die Modellierung des FT als GSPN sehr viele *vanishing* Markierungen erzeugt, die bei der Generierung der zum GSPN gehörigen Markovkette jedoch aus dem Zustandsraum eliminiert werden. Die Generierung dieser *vanishing* Markierungen läßt sich vollständig vermeiden, wenn man den FT statt als GSPN als SRN mit der Gewinnfunktion (*Reward*)  $f$  darstellt. Diese Funktion ordnet jeder Markierung  $m$  des SRN den Wert 1 zu, falls die Markierung alle Bedingungen für das Auftreten des TE erfüllt, den Wert 0 sonst. Für das in Abbildung 1 dargestellte System lautet die Gewinnfunktion, welche direkt die Wahrscheinlichkeit des TE ausdrückt:

$$f(m) = [\#(B1:s1, m) \wedge \#(B2:s1, m)] \vee [\#(B3:s2, m) \wedge (F_{2/3}(\#(B4:s2, m), \#(B5:s2, m), \#(B6:s2, m)) \vee (\#(B6:s2, m) \wedge \#(B7:s1, m) \wedge \#(B8:s2, m)))]$$

Die Funktion  $\#(p, m)$  sei 1, wenn die Stelle  $p$  in der aktuellen Markierung  $m$  Marken enthält, 0 sonst.  $F_{2/3}$  sei die zum 2-aus-3-Gatter G4 gehörige Funktion. Wie aus Tab. 1 ersichtlich, kann die Anzahl der *vanishing* Markierungen die Anzahl der *tangible* Markierungen abhängig vom betrachteten System zum Teil deutlich übersteigen. Wird der FT jedoch als Funktion dargestellt, werden nur die durch die Modellierung der *Common Cause Failure* bedingten *vanishing* Markierungen erzeugt. Für die Auswertung wurden die Komponenten B3, B7, B8 und B9 identisch zu B1 und B4, B5 und B6 identisch zu B2 definiert.

(Teil-)Baum	Stellen im GSPN	Transitionen im GSPN	<i>tangible</i> Markierungen des GSPN	<i>vanishing</i> Markierungen des GSPN
G0 (GSPN)	54	58	59049	170511
(Funktion)	36	46	59049	32825
M1 (GSPN)	9	10	9	5
(Funktion)	6	8	9	—
M2 (GSPN)	42	46	6561	11043
(Funktion)	30	38	6561	3626
M3 (GSPN)	36	40	2187	3118
(Funktion)	27	34	2187	1212
MB (GSPN)	3	4	3	—
(Funktion)	3	4	3	—

Tabelle1. Größe der GSPN zum Modell aus Abb. 1

**Reduktion der *tangible* Markierungen** Die Verringerung der Anzahl der *tangible* Markierungen des erzeugten GSPN/SRN läßt sich durch die Modularisierung des FT erreichen. Das Verfahren der Modularisierung stammt aus der klassischen FTA. Der FT wird hierbei dergestalt in Teilbäume zerlegt,

daß Blätter und Gatter eines Teilbaums keine Verbindungen zu Gattern außerhalb dieses Teilbaums besitzen. Einen Teilbaum, der diese Bedingung erfüllt, nennt man *Modul* [5]. Häufig entsprechen Module größeren Teilsystemen oder Untereinheiten des realen Systems. In Abb. 1 ist der Teilbaum mit der Wurzel G1 ein Modul, der Teilbaum T mit der Wurzel G5 jedoch nicht, da sein Blatt B6 eine Verbindung zu Gatter G4 hat, welches nicht in T ist. Die Eintrittswahrscheinlichkeit des Wurzelereignisses eines Moduls läßt sich nun unabhängig vom Rest des Baumes bestimmen. Anschließend kann das Modul im ursprünglichen FT wie ein Blatt behandelt werden.

Ein erweitertes FT-Modell läßt sich ebenso wie ein klassischer FT in Module zerlegen. Hierbei muß allerdings zusätzlich berücksichtigt werden, daß keine Komponente innerhalb eines Moduls von einer Komponente außerhalb des Moduls stochastisch abhängig sein darf. Die Module M1 und M3 mit den Wurzeln G1 und G3 aus Abb. 1 erfüllen auch diese Bedingung. Enthält der FT, der entsteht, wenn alle Module durch einfache Blätter ersetzt werden, keinerlei stochastische Abhängigkeiten zwischen seinen Blättern mehr, so kann die Lösung der aus den Modulen erzeugten GSPN/SRN direkt in den FT eingesetzt werden und die Wahrscheinlichkeit des TE mit Methoden der klassischen FTA bestimmt werden. Aus Tab. 1 ist ersichtlich, daß bei der Zerlegung des FT aus Abb. 1 in die Module M1 und M2 bzw. M1, M3 und das Modul MB bestehend aus der Komponente B3 jeweils über 85% bzw. 95% weniger Markierungen betrachtet werden müssen.

## 4 Ausblick

In der klassischen FTA wird häufig die Methode der *Truncation* angewendet, um für große Systeme in angemessener Zeit eine Näherungslösung bestimmen zu können. Ähnliche Methoden sind bei der Analyse erweiterter FT ebenfalls denkbar, um die Anzahl der *tangible* Markierungen der erzeugten GSPN bzw. SRN weiter zu verringern.

In diesem Beitrag wurde die Modellierung mit erweiterten FT vorgestellt, deren Auswertung auf der Basis von GSPN und SRN geschieht. Es stellt sich die Frage, inwiefern die Ergebnisse bei der Analyse eines Modells mit diesem Verfahren mit Ergebnissen, die durch klassische FTA gewonnen wurden übereinstimmen. Für Modelle, die sich auch mit der klassischen FTA lösen lassen, wird gefordert, daß die Ergebnisse der Auswertung bei beiden Verfahren identisch sind. Für die beiden Spezialfälle eines Systems mit entweder ausschließlich reparierbaren oder ausschließlich nicht reparierbaren Komponenten konnte dies für die Transformation in GSPN allgemein nachgewiesen werden.

## Literatur

1. M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons, Chichester, 1995.
2. S. Allmaier S. Dalibor. PANDA — **Petri net ANalysis and Design Assistant**. in *Tools Descriptions, 9th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, S. 58–60, 1997.
3. G. Ciardo, A. Blakemore, P. F. Chimento, J. K. Muppala, K. S. Trivedi. **Automated Generation and Analysis of Markov Reward Models Using Stochastic Reward Nets**. in C. Meyer R. J. Plemmons (Hrsg.), *Linear Algebra, Markov Chains, and Queueing Models*, S. 145–191. Springer Verlag, Heidelberg, 1993.
4. B. Grams. **Entwurf und Implementierung von anwendungsbasierten Methoden zur Dependability-Analyse basierend auf stochastischen Petri-Netzen**. Master's thesis, Institut für Mathematische Maschinen und Datenverarbeitung der Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Informatik III (Rechnerstrukturen), April 1995.
5. T. Kohda, E. J. Henley, K. Inoue, **Finding Modules in Fault Trees**, in *IEEE Transactions on Reliability* **38** (1989), 2, S. 165–176.
6. W. S. Lee, D. L. Grosh, F. A. Tillman, C. H. Lie, **Fault Tree Analysis, Methods, and Applications — A Review**, in *IEEE Transactions on Reliability* **R-34** (1985), 3, S. 194–203.
7. M. Malhotra K. S. Trivedi, **Dependability Modeling Using Petri Nets**, in *IEEE Transactions on Reliability* **44** (1995), S. 428–440.
8. N. H. Roberts, W. E. Vesely, D. F. Haasl, F. F. Goldberg. *Fault Tree Handbook*. Number NUREG-0492. System and Reliability Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, D.C. 20555, 1981.
9. B. Specker. **Evaluation of Fault-Tolerant Systems Using Stochastic Petri Nets and Fault Trees**. in *European Simulation Symposium*, S. 83–87, 1992.
10. S. K. Yang T. S. Liu, **Failure Analysis for an Airbag Inflator by Petri Nets**, in *Quality and Reliability Engineering International* **13** (1997), S. 139–151.