Einführung in stochastische Petrinetze

Proseminar Technische Informatik SoSe 11

Christoph Krüger

10. Juni 2011

Zusammenfassung

Stochastische Petrinetze sind ein weit verbreitetes und gutes Mittel, um komplizierte und massiv nebenläufige Systeme zu analysieren. Sie stellen ein leicht nachvollziehbares graphisches Modell, welches einen guten Überblick über das zu analysierende System bietet, sowie eine stabile mathematische Grundlage für die weiteren Berechnungen zur Verfügung. Eine Reihe von Tools erleichtert die Erstellung und Analyse dieses Modells. All dies führt dazu, dass man mit stochastischen Petrinetzen frühzeitig mögliche Probleme des Systems und vorraussichtlich zu erwartenden Ressourcenbedarf abschätzen, sowie Gewissheit über die Erfüllung bestimmter Anforderungen erlangen kann.

1 Einleitung

Normale Petrinetze sind eine fantastische Möglichkeit, um Algorithmen und Systeme zu spezifizieren und einzelne Eigenschaften, vor Allem nebenläufiger Algorithmen, beispielsweise die Abwesenheit von Deadlocks oder gegenseitigen Ausschluss, zu überprüfen.

Auf gleiche Art und Weise kann man mittels verschiedener Varianten stochastischer Petrinetze so gut wie jedes System analysieren oder zumindest auf einfache Art und Weise simulieren, um beispielsweise Ressourcenverbrauch, Wartezeiten, Durchsatz und Ausfallwahrscheinlichkeiten in komplizierten Systemen oder die Menge bestimmter Redundanzen, die benötigt werden, um eine gewisse Verfügbarkeit zu erhalten, berechnen zu können, ehe das System gebaut wird. Diese Arbeit soll einen Überblick über die Möglichkeiten von Petrinetzen geben und eine Idee des mathematischen Modells bieten, ohne den Leser mit der ganzen Komplexität des Themas zu erschlagen.

Im Groben ist die Arbeit wie folgt aufgebaut: Im ersten Teil werde ich auf Markovketten in stetiger Zeit eingehen, da diese einen Teil der mathematischen Grundlage von stochastischen Petrinetzen bilden und später benötigt werden, um gewisse Analysen auf Stochastischen Petrinetzen durchzuführen. Es handelt sich lediglich um eine kurzen Überblick der Idee von Markovketten, die wir für das Verständnis dieser Arbeit benötigen. Für ein tieferes Verständnis sei auf die angegebene Literatur verwiesen. Wenn sie sich bereits mit Markovketten auskennen, können sie diesen Teil getrost überspringen.

Im zweiten Teil werde ich versuchen, ihnen das den Stochastischen Petrinetzen zugrunde liegende Modell des Petrinetzes nahezubringen. Er erklärt das graphische Modell und erläutert Eigenschaften, auf die im Folgenden zurückgegriffen werden. Während bei dem ersten Abschnitt die Mathematischen Grundlagen auch erst bei Bedarf nachgelesen werden können, ist das Verständnis dieses Abschnittes unabdingbar, um die folgenden Abschnitte zu verstehen. Sollten sie sich jedoch bereits gut mit Petrinetzen auskennen und sich nur für die Erweiterung zu den Stochastischen Petrinetzen interessieren, können sie auch diesen Abschnitt überspringen.

Der dritte Abschnitt bildet den eigentlichen Kern der Arbeit. Er stellt stochastische Petrinetze vor und erläutert, wie man mit ihnen ein System durch bestimmte Berechnungen analysieren kann.

Im vierten Abschnitt werde ich einige Tools zeigen, die die Erstellung und Analyse von Stochastischen Petrinetzen erleichtern sollen. Es handelt sich nicht um eine Einführung, sondern lediglich um Hinweise, wann und warum man sich mit ihnen beschäftigen sollte.

2 Markovketten

Markovketten sollen hier kurz vorgestellt werden, da wir später einen Teil der Stochastischen Petrinetze darauf zurückführen werden. Sie dienen dazu Wahrscheinlichkeiten für das Eintreten zukünftiger Ereignisse in gedächtnislosen Systemen zu berechnen.

Markovketten modellieren Systeme, die zufällig von einem Zustand in einen anderen übergehen. Man unterscheidet zwischen Markovketten in stetiger Zeit und Markovketten in diskreter Zeit. Hier soll es nur um Markovketten in stetiger Zeit gehen, da wir nur diese für das Verständnis dieser Arbeit benötigen. Markovketten haben immer einen diskreten Zustandsraum, was sie von Markovprozessen abhebt. Zwischen Zuständen (modelliert als Knoten eines Graphen) wird sprunghaft gewechselt. Sie sind gedächtnislos, was im stetigen Fall bedeutet, dass Zustandsübergänge (modelliert als Pfeile) nur von der betrachteten Zeitspanne und dem aktuellen Zustand und nicht von der insgesamt verstrichenen Zeit, abhängig sind.

Die Übergangswahrscheinlichkeiten lassen sich bei stetigen Markovketten ebenso in einer Matrix abbilden, wie es bei diskreten Markovketten möglich ist. Der Unterschied besteht darin, dass die Übergangswahrscheinlichkeiten und damit auch die Matrix in Abhängigkeit von der Zeit angegeben werden. [**?**, Abs. 17.3]

3 Allgemeine Grundlagen von Petrinetzen

In diesem Abschnitt sollen grundlegende Elemente und Eigenschaften von Stochastischen Petrinetzen vorgestellt werden, die sie sich mit den normalen Petrinetzen teilen. Dies wurde in einen eigenen Abschnitt ausgelagert, damit diejenigen, die sich bereits mit Petrinetzen auskennen, die für sie relevanten Informationen leichter lokalisieren können.

Petrinetze sind deswegen so gut für die Modellierung von Systemen geeignet, weil sie gleichzeitig eine leicht zu verstehende graphische Darstellung, in der Fehler in der Modellierung schnell erkannt werden können, und eine stabile mathematische Grundlage für die Berechnungen, bieten.

3.1 Aufbau

Nehmen wir an, wir wollen ein Cluster aus 2 Webservern errichten. Jeder dieser Webserver kann eingehende Anfragen bearbeiten solange er stabil läuft. Leider können diese natürlich auch manchmal ausfallen. Nun wollen wir dies mit einem einfachen Petrinetz modellieren (siehe 1), wobei wir ein sicheres (siehe 3.2) Netz modellieren, um später nicht den Rahmen der Arbeit zu sprengen.

Petrinetze bestehen aus aktiven und passiven Komponenten.

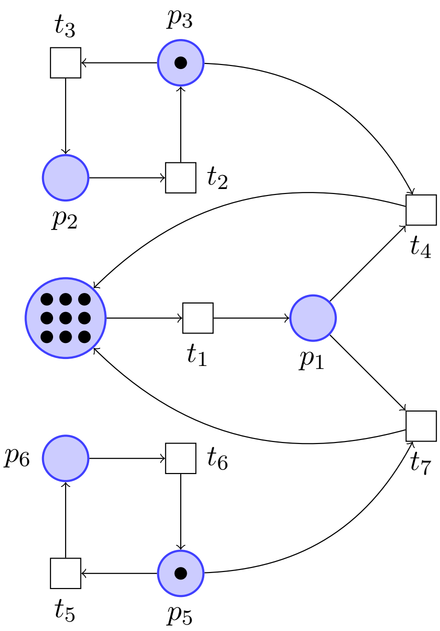
Passive Komponenten sind die sogenannten Stellen oder Plätze, dargestellt durch leere Kreise. Diese stellen ein bestimmtes mögliches Objekt in einem bestimmten Zustand, also zum Beispiel einen ausgefallenen Webserver, dar. Ob und wieviele solche Objekte es gibt, wird durch die Anzahl der Marken auf einer Stelle dargestellt. Jede Marke repräsentiert ein Objekt und wird soweit möglich durch einen schwarzen Punkt in einer Stelle dargestellt. Sollten es zu viele Marken werden, kann man natürlich auch einfach die Zahl darin notieren.

Aktive Komponenten sind die sogenannten Transitionen. Sie werden durch Rechtecke dargestellt und repräsentieren eine Handlung, die im System vollzogen wird. Also in unserem Beispiel etwa ,,ein Webserver fällt aus” oder ,,eine Anfrage wird verarbeitet”.

Aktive und passive Komponenten werden durch Pfeile verbunden. Diese Repräsentieren Marken, die beim Schalten (teils auch ,,Feuern” genannt) der Transition verbraucht (Pfeile von Stellen zu Transitionen) oder erzeugt (Pfeile von Transitionen zu Stellen) werden. Pfeile können mit Zahlen versehen werden, wenn beim Schalten mehrere Marken an einer Stelle verbraucht oder erzeugt werden sollen.

Zusammen bildet dies bereits ein Petrinetz, welches Mathematisch als 5-Tupel

$$PN = \{P, T, F, W, M_0\}$$



t_1 = Anfrage geht ein

p_1 = Wartende Anfragen

t_4 = Anfrage wird bearbeitet

t_7 = Anfrage wird bearbeitet

p_3 = Laufender Webserserver 1

t_3 = Webserver 1 stürzt ab

p_2 = Abgestürzter Webserserver 1

t_2 = Webserver 1 wird repariert

p_5 = Laufender Webserserver 2

t_5 = Webserver 2 stürzt ab

p_6 = Abgestürzter Webserserver 2

t_6 = Webserver 2 wird repariert

Abbildung 1: Ein Cluster aus 3 Webservern als Petrnetz

mit der Menge der Stellen $P = \{p_1, \ldots, p_n\}$, der Transitionen $T = \{t_1, \ldots, t_n\}$, der Pfeile $F \subseteq P \times T \cup T \times P$, der Anzahl der Marken für jeden Pfeil $W\colon F \to \mathbb{N}$ und der Anzahl der Marken, die zu Beginn auf jeder Stelle liegen $M_0\colon P \to \mathbb{N}$, dargestellt wird. [BK02, Kap. 5] [Haa02, Kap. 3,4] [MBC$^+$95] [Spr04] [Bau97]

3.2 Eigenschaften von Petrinetzen

Petrinetze, beziehungsweise ihre Elemente, besitzen gewisse Eigenschaften, die für die Analyse von Stochastischen Petrinetzen wichtig sind. Hierzu gehören

Vorgänger: All jene Plätze, die Marken enthalten müssen, damit t schalten kann, nennen wir Vorgänger von t.

Nachfolger: Diejenigen Plätze, die Marken erhalten, wenn t schaltet, bezeichnen wir als dessen Nachfolger.

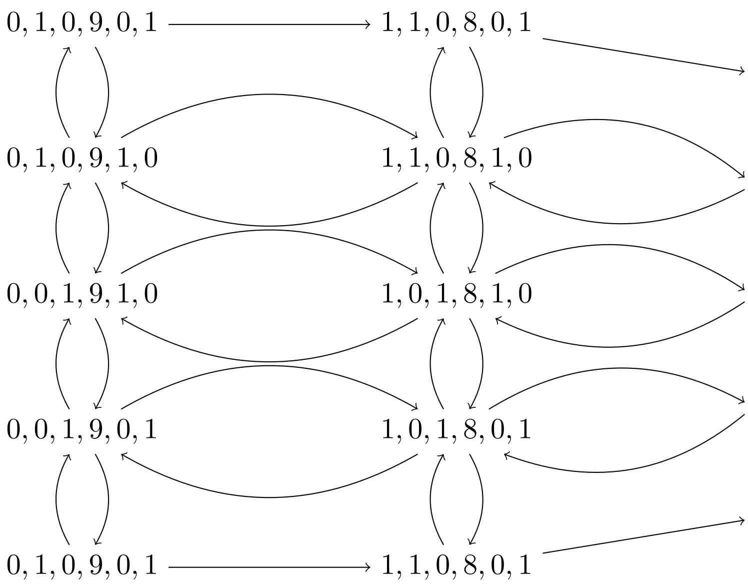


Abbildung 2: Aussschnitt des Erreichbarkeitsgraphen unseres Beispielsystems

Aktiviert: Wir bezeichnen eine Transition als aktiviert, wenn all ihre Vorgänger über ausreichend Marken verfügen, damit sie Schalten kann.

Sicherheit: Wir bezeichnen einen Zustand als n-sicher, wenn er nie mehr als n Marken erhalten kann. Ebenso ist ein Netz n-sicher, wenn alle Plätze mindestens n-sicher sind.

Erreichbar: Ein Zustand M_n heißt erreichbar, wenn es eine Schaltfolge gibt, die M_0 in M_n überführt. Überführt eine Transition t den Zustand M_i in den Zustand M_{i+1} schreiben wir $M_i[t > M_{i+1}$. Dies ist die Grundlage für den später benötigten Erreichbarkeitsgraphen, bei dem jeder erreichbare Zustand einen Knoten und jede Transition dazwischen eine Kante bildet (Siehe 2). Wenn das Netz nicht sicher ist, ist dieser endlos. In der Algorithmik wird er übrigens benutzt, um beispielsweise die Abwesenheit von Deadlocks oder den gegenseitigen Ausschluss zu zeigen.

[Bau97]

4 Stochastische Petrinetze

Wir wollen Systeme über einen zeitlichen Bereich betrachten und es ist unzureichend anzunehmen, dass jede Transition sofort schaltet, sobald sie aktiviert ist. In unserem Beispiel vergeht etwas Zeit, ehe eine Anfrage fertig bearbeitet ist, es gehen nicht ständig neue Anfragen ein und auch unsere Server fallen glücklicherweise nicht ständig aus. Um dies zu modellieren, benötigen wir die stochastischen Petrinetze ein. Diese gehen davon aus, dass jede Transition eine gewisse Zeit benötigt, um zu schalten. Hierbei müssen wir unterscheiden zwischen Netzen, die annehmen, dass eine aktivierte Transition sich sofort die benötigten Ressourcen(Marken) sichert (preselection) und solchen, die davon ausgehen, dass sie erst beim Schalten die Marken sichern und verbrauchen (race). Wir gehen hier von race Netzen aus, da ein Server immer noch ausfallen kann, wenn er eigentlich eine Anfrage bearbeitet.

4.1 Aufbau

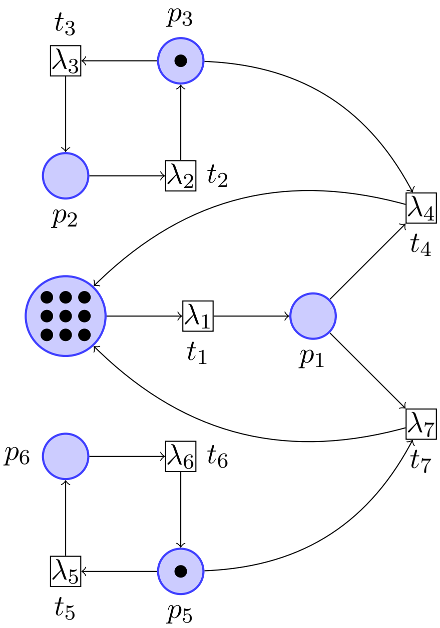
Einem stochastische Petrinetz liegt ein Petrinetz, wie es in 3.1 beschrieben ist, zu Grunde. Jede Transition wird dabei um eine Feuerrate λ ergänzt (siehe Abb. 3). Wir erhalten also ein 6-Tupel

$$SPN = \{P, T, F, W, M_0, \Lambda\}$$

mit $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ der Menge der einzelnen Feuerraten. Diese entsprechen dem Kehrwert der durchschnittlichen Zeit, die diese Transition benötigt. Die Zeit, die die Transition dann wirklich benötigt, wenn sie aktiviert wird, wird zufällig über die Exponentialfunktion $e^{-x\lambda}$ bestimmt.

4.2 Berechnungsbeispiele

Nehmen wir nun an, wir wollen in unserem Modell die Wahrscheinlichkeit berechnen, dass alle 3 Server gleichzeitig ausgefallen sind. Ist das Netz sicher, so können wir hierfür das Stochastische Petrinetz auf eine stetige Markovkette zurückführen, indem wir den Erreichbarkeitsgraphen bilden und jede Kante von M_i nach M_j mit der Rate λ_k der



t_1 = Anfrage geht ein

p_1 = Wartende Anfragen

t_4 = Anfrage wird bearbeitet

t_7 = Anfrage wird bearbeitet

p_3 = Laufender Webserserver 1

t_3 = Webserver 1 stürzt ab

p_2 = Abgestürzter Webserserver 1

t_2 = Webserver 1 wird repariert

p_5 = Laufender Webserserver 2

t_5 = Webserver 2 stürzt ab

p_6 = Abgestürzter Webserserver 2

t_6 = Webserver 2 wird repariert

Abbildung 3: Abb. 1 um Feuerraten ergänzt

Transition, die M_i in M_j überführt, versehen. Es geht auch mit einigen unsicheren Netzen, wenn der Erreichbarkeitsgraph abzählbar unendlich ist. Dies zu zeigen würde jedoch den Rahmen dieser Arbeit sprengen. Jeder Knoten bildet hierbei einen Zustand der Markovkette. Diese hat die Generatormatrix Q mit:

$$q_{i,j} = \begin{cases} \lambda_k & i \neq j \\ -\sum\limits_{n \neq i} q_{i,n} & i = j \end{cases} \quad \text{mit} M_i[t_k > M_j$$

Hier sieht man gut, wie riesig diese Matrix bereits für unser kleines Beispiel wird und warum man Tools für die Analysierung benutzt. Zur Verdeutlichung der Form sei hier ein kleiner Ausschnitt gegeben:

$$Q = \begin{array}{c} \\ \{0,0,1,9,1,0\} \\ \{0,0,1,9,0,1\} \\ \{0,1,0,9,1,0\} \\ \dots \end{array} \begin{array}{c} \begin{array}{ccccc} \{0,0,1,9,1,0\} & \{0,0,1,9,0,1\} & \{0,1,0,9,1,0\} & \{1,0,1,8,1,0\} & \dots \end{array} \\ \begin{pmatrix} -(\lambda_1+\lambda_5+\lambda_3) & \lambda_5 & \lambda_3 & \lambda_1 & \dots \\ \lambda_6 & -(\lambda_1+\lambda_3+\lambda_6) & \lambda_3 & \lambda_1 & \dots \\ \lambda_2 & 0 & -(\lambda_2+\lambda_1+\lambda_5) & \dots & \\ \dots & & & & \end{pmatrix} \end{array}$$

Aus dieser können wir nun den Vektor π berechnen, der angibt, wie groß die Wahrscheinlichkeit für einen bestimmten Zustand ist, indem wir die Formeln

$$\pi Q = 0 \qquad \text{und} \qquad \sum \pi_i = 1$$

lösen. Addieren wir nun die Wahrscheinlichkeiten all jener Zustände, in denen alle Server ausgefallen sind (in unserem Beispiel $(0,1,0,9,0,1)\dots(1,1,0,8,0,1)$) so erhalten wir die Wahrscheinlichkeit für einen Totalausfall. Desweiteren interessiert uns natürlich, wieviele Token in der Warteschlange zu erwarten sind. Dies berechnet sich über

$$E_i = \sum_{n=1}^{\infty}(n \cdot P[B(p_i, n)]$$

in unserem Beispiel natürlich mit $i = 1$. Weiterhin würde es uns natürlich interessieren, wieviele Anfragen pro Zeit nun tatsächlich bearbeitet werden. Hierzu bilden wir die Mengen A_j aller Zustände, in denen t_j aktiviert ist. In unserem Beispiel interessiert uns $j = 4$ und $j = 7$, um jeweils den Durchsatz des ersten bzw des zweiten Webservers zu erhalten.

$$d_j = \sum_{M_i \in A_j} \pi_i \lambda_j$$

[BK02, S. 134 ff]

4.3 Allgemeine Stochastische Petrinetze

Häufig reichen exponentielle Verteilungen nicht aus, um ein System ausreichend zu simulieren. Beispielsweise, wenn man Vorgänge hat, die eine konstante Zeit benötigen, gewisse Vorgänge immer bevorzugt ablaufen sollen oder ähnliches. Viele Webdienste haben beispielsweise Premiumuser, die bevorzugt behandelt werden. Hier interessiert natürlich in erster Linie nicht die gesamte Länge der Warteschlange, sondern nur die Länge der Premiumwarteschlange. In solchen Fällen greift man auf die allgemeinen stochastischen Petrinetze (GSPN) zurück. Diese unterteilen die Menge der Transitionen in verzögerte Transitionen und unverzögerte Transitionen. Unverzögerte Transitionen hebt man dabei von verzögerten graphisch ab, indem man sie als Strich darstellt und mit einem großen statt einem kleinen t benennt.

Desweiteren verfügen diese Auf Test und Verbotskanten, die nur auf die Anwesenheit von Marken an einer bestimmten Stelle testen, ohne diese zu verbrauchen, und dann die entsprechenden Transitionen gegebenenfalls verbieten. Auch hier ist wieder die Darstellung von Premiumusern eine beispielhafte Anwendung, aber es lassen sich auch andere Anwendungsfälle denken. So ist es etwa möglich, in einem redundanten System das Durchführen von geplanten Downtimes zu verbieten, solange ein Teil des Systems ausgefallen ist. Beide Arten werden durch eine Kante mit einem kleinen Kreis am Ende dargestellt, wobei dieser bei Verbotskanten ausgefüllt ist. [MBC+95] [Abe08, S. 69]

5 Tools

Da man die Berechnungen nicht immer von Hand durchlaufen lassen will und diese irgendwann auch an ihre Grenzen stoßen, gibt es Tools, die einem ermöglichen, dass man selbst nur noch das Stochastische Petrinetz zeichnet und die gewünschten Raten angibt. Alles weitere, wie das Berechnen des Erreichbarkeitsgraphen, die Rückführung auf eine Markovkette, sowie die Simulation des Ablaufes gemäß der gegebenen Regeln („ das Tokenspiel spielen"), übernehmen die Programme dann automatisch.

Zwei Vertreter dieser Programme wären zum Beispiel Timenet und GreatSPN. Beide sind für Lehrzwecke frei verfügbar und können mit Bestätigung der Universität bezogen werden. Neben den oben vorgestellten Varianten von stochastischen Petrinetzen können sie auch (erweiterte) deterministische stochastische Petrinetze, also solche, die nicht nur exponentielle und unverzögerte, sondern beliebige Verteilungen benutzen können, erzeugen und analysieren. [ZFGH, Kap. 2] [Zim01]

6 Schluss

Insgesamt ist durch diese Arbeit hoffentlich deutlich geworden, welch mächtige Hilfsmittel Petrinetze und besonders stochastische Petrinetze sind. Natürliche wurden die Beispiele in dieser Arbeit relativ klein gehalten, damit die Netze und Graphen nicht zu groß werden, doch sollte trotzdem deutlich geworden sein, dass sich diese Mittel auch auf sehr viel größere und kompliziertere Systeme anwenden lassen. Ich hoffe, es wurde ein Grundverständniss für Petrinetze und ihre Fähigkeiten zur Modellierung und Analyse vermittelt und eine Anregung für die weitere Beschäftigung mit dem Thema und seinen mathematischen Hintergründen, die hier nicht in vollem Umfang vermittelt werden konnten, gegeben.

Literatur

[Abe08] Marcus Abele. *Modellierung und Bewertung hochzuverlässiger Energiebordnetz-Architekturen für sicherheitsrelevante Verbraucher in Kraftfahrzeugen.* Kassel University Press, 3 2008.

[Bau97] Bernd Baumgarten. *Petri-Netze: Grundlagen und Anwendungen (German Edition).* Spektrum Akademischer Verlag, 2. aufl. edition, 1 1997.

[BK02] Falko Bause and Pieter S. Kritzinger. *Stochastic Petri Nets.* Friedrich Vieweg & Sohn Verlag, 8 2002.

[Haa02] Peter J. Haas. *Stochastic Petri Nets.* Springer, 1 edition, 6 2002.

[MBC+95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* John Wiley & Sons, 1 edition, 11 1995.

[Spr04] *Applications and Theory of Petri Nets 2004: 25th International Conference, ICATPN 2004, Bologna, Italy, June 21-25, 2004, Proceedings (Lecture Notes in Computer Science).* Springer, 1 edition, 8 2004.

[ZFGH] Armin Zimmermann, Jörn Freiheit, Reinhard German, and Günter Hommel. Petri net modelling and performability evaluation with timenet 3.0. In Boudewijn Haverkort, Henrik Bohnenkamp, and Connie Smith, editors, *Computer Performance Evaluation.Modelling Techniques and Tools*, volume 1786 of *Lecture Notes in Computer Science*, pages 188–202. Springer Berlin / Heidelberg.

[Zim01] Armin Zimmermann. Timenet 3.0 user manual, a software tool for the performability evaluation with stochastic petri nets. 2001.