

Petrietze und GPSS/H

Robert Siegfried

Hochschule Mittweida (FH)
Seminararbeit Simulation

11. Januar 2006

Agenda

- 1 Einleitung
- 2 Petrietze
- 3 GPSS/H
- 4 Fazit

Agenda

- 1 Einleitung
- 2 Petrietze
- 3 GPSS/H
- 4 Fazit

Themen

- Modellierung von dynamischen Systemen
- Formale Modellierung von nebenläufigen Systemen
- Simulation von nebenläufigen Systemen
- Simulation von Warteschlangensystemen
- Präsentation der Simulationssoftware GPSS/H

Agenda

- 1 Einleitung
- 2 Petrinetze**
- 3 GPSS/H
- 4 Fazit

Überblick

- 1962 von Carl Adam Petri entwickelt (Dissertation „Kommunikation mit Automaten“, TU Darmstadt).
- Formalismus zur Modellierung, Simulation und Analyse von Systemen mit folgenden Eigenschaften
 - Nebenläufigkeit
 - Synchronisation
 - Nichtdeterminismus
- Graphisch anschaulich, zugleich gut formalisierbar und sehr ausdrucksstark.
- Vielfältig erweiterbar.

Definition – informal

Ein Petrietz ist ein gerichteter Graph bestehend aus

- Stellen, Plätze (Knoten)
- Transitionen (Knoten)
- Verbindungen (Kanten)
- Markierungen

Zusätzlich gibt es i. d. R. einen beschreibenden Teil, in dem weitere Details spezifiziert werden können.

Grafische Notation

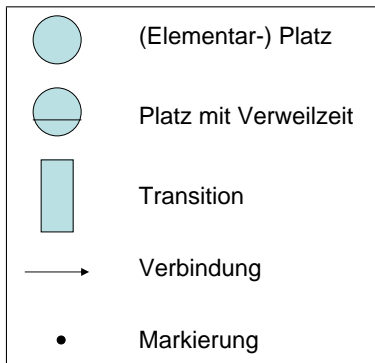


Abbildung: Grafische Notation von Petrinetzen

Beispiel

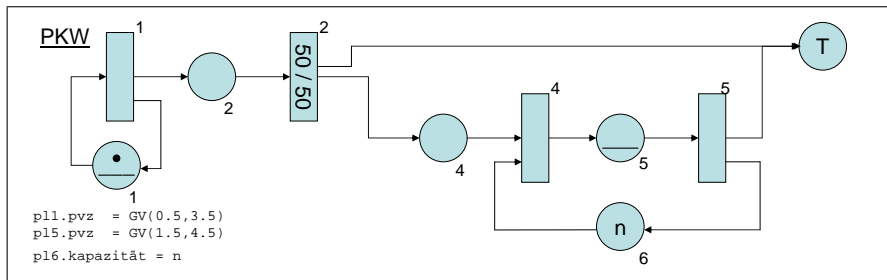


Abbildung: Beispiel (Zollabfertigung von PKW)

Verhalten

- Vorgänger und Nachfolger von jeder Stelle ist eine Transition.
- Vorgänger und Nachfolger jeder Transition ist ein Platz. Jede Transition besitzt mindestens einen Vorgänger und Nachfolger.
- Um das Verhalten des Petrietzes zu schreiben, werden Marken verwendet. Lediglich Plätze können markiert werden.
- Eine Transition kann nur schalten, wenn auf allen vorhergehenden Plätzen eine Markierung vorhanden ist.
- Beim Schalten einer Transition werden Marken auf den vorhergehenden Plätzen vernichtet und auf allen nachfolgenden Plätzen erzeugt.

Beispiel

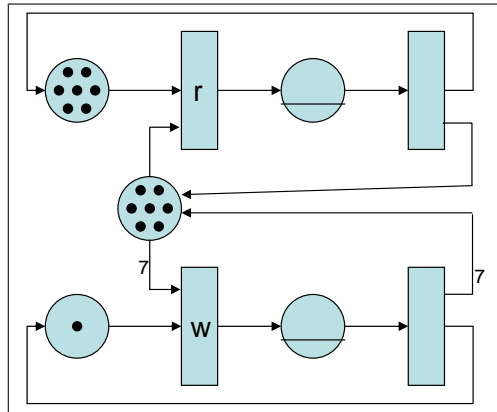


Abbildung: Konkurrierende Lese-/Schreibzugriffe (r/w)

Beispiel

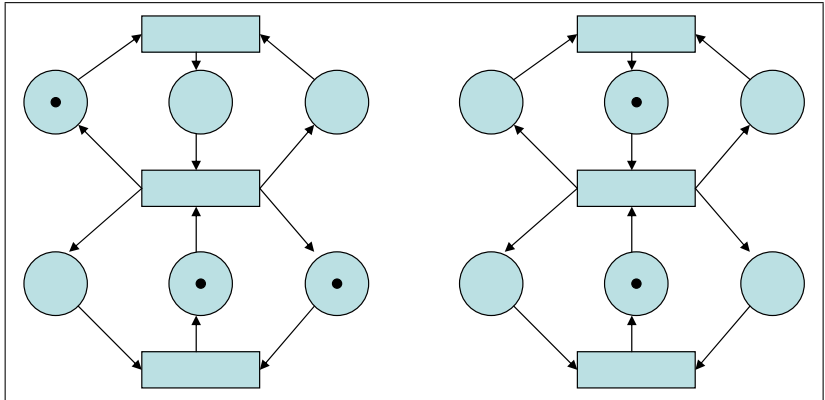


Abbildung: Petrietze mit und ohne Deadlock

Formale Definition

Ein Petrietz ist ein 6-Tupel (S, T, F, K, W, m_0) mit

- $S = \{s_1, \dots, s_n\}$ Menge der Stellen
- $T = \{t_1, \dots, t_m\}$ Menge der Transitionen
- $F \subseteq (S \times T) \cup (T \times S)$ Flussrelation
- $K : S \rightarrow \mathbb{N} \cup \{\infty\}$ Kapazitäten der Stellen
- $W : F \rightarrow \mathbb{N}$ Gewichte der Verbindungen
- $m_0 \in \mathbb{N}^n$ Startmarkierung

Nahezu beliebig erweiterbar (z. B. um Plätze mit Verweilzeiten).

Schaltbereitschaft

- $Z := \mathbb{N} \times \mathbb{N}$ Zeitpunkt (eines Ereignisses)
- $M : S \times Z \rightarrow \mathbb{N}$ Zustand des Petrinetzes (Markierung)
- $pre(t) := \{s \in S : (s, t) \in F\}$
Vorgänger einer Transition $t \in T$ (engl. predecessor)
- $succ(t) := \{s \in S : (t, s) \in F\}$
Nachfolger einer Transition $t \in T$ (engl. successor)

Eine Transition $t \in T$ heißt *aktiviert* (oder schaltbereit), gdw.

- 1 $\forall s \in pre(t) : M(s) \geq W(s, t)$
- 2 $\forall s \in succ(t) \setminus pre(t) : K(s) \geq M(s) + W(t, s)$
- 3 $\forall s \in pre(t) \cap succ(t) : K(s) \geq M(s) + W(t, s) - W(s, t)$

Schaltvorgang

Eine schaltbereite Transition $t \in T$ kann schalten. Die Markierungen werden dann wie folgt neu berechnet:

- $s \in pre(t) \wedge s \notin succ(t) : M'(s) = M(s) - W(s, t)$
- $s \notin pre(t) \wedge s \in succ(t) : M'(s) = M(s) + W(t, s)$
- $s \in pre(t) \wedge s \in succ(t) : M'(s) = M(s) - W(s, t) + W(t, s)$

Befinden sich mehrere aktivierte Transitionen in Konflikt (d. h. das Schalten einer Transition führt dazu, dass eine andere nicht mehr schalten kann), so ist über ein *Transitionsauswahlverfahren* die Schaltreihenfolge festzulegen.

Beispiel

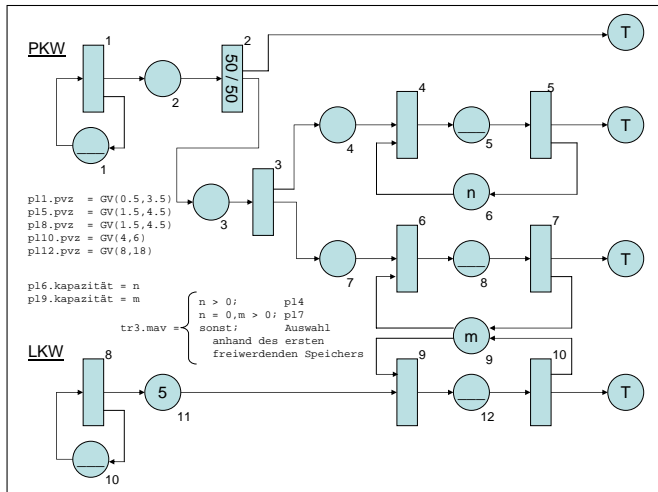


Abbildung: Beispiel (Zollabfertigung von PKW und LKW)

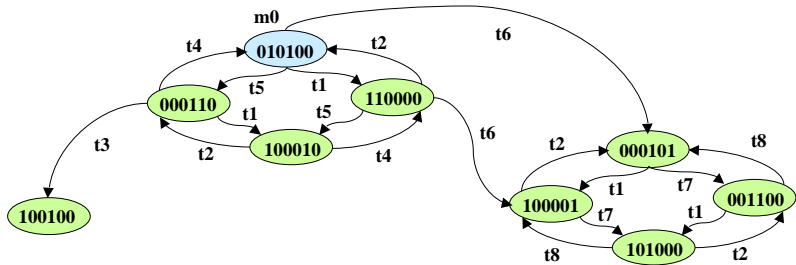
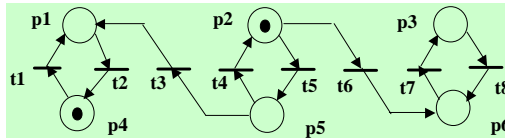
Standardisierung

- ISO/IEC 15909-1 Software and Systems Engineering – High-level Petri Nets – Concepts, Definitions and Graphical Notation
- ISO/IEC 15909-2 Software and Systems Engineering – High-level Petri Nets Part 2: Transfer Format
- Teil 3 soll Ergänzungen zu Teil 1 enthalten, u. A.:
 - Modularisierung
 - Hierarchien in sog. *Coloured Petri nets*
 - Erweiterung des Zeitbegriffs
 - Stochastische Petrietze

Lebendigkeit

- Eine Markierung $m \in M$ heißt *tot*, falls keine Transition $t \in T$ schalten kann bei m .
- Eine Transition $t \in T$ heißt *tot* bei der Markierung $m \in M$, wenn von m aus keine Markierung erreichbar ist, bei der t schalten kann.
- Eine Transition $t \in T$ heißt *lebendig* bei der Markierung $m \in M$, wenn t bei keiner von m aus erreichbaren Markierung tot ist.
- Eine Markierung $m \in M$ heißt *lebendig*, wenn alle Transitionen $t \in T$ bei m lebendig sind.
- Ein Petri-Netz heißt *lebendig*, wenn die Anfangsmarkierung m_0 lebendig ist.

Erreichbarkeitsgraph



Totale Verklemmung

Partielle Verklemmung

Agenda

- 1 Einleitung
- 2 Petrietze
- 3 GPSS/H**
- 4 Fazit

Überblick

- General Purpose Simulation System
- Entwickelt in den 60ern.
- Nach heutigem Empfinden „ungewohnte“ Syntax.
- Dafür aber schnell und flexibel.

Wesentliche Elemente

- Statische Elemente
 - Einrichtungen
 - Speicher
- Dynamische Elemente
 - Aktivatoren
- Statistische Elemente
 - Warteschlangen
- Blöcke (Aktionen/Ereignisse mit Auswirkungen auf die Aktivatoren)
- Steueranweisungen

Einrichtungen

- Modellierung von Ein-Bedienstationen.
- Mögliche Zustände: belegt / frei

Verwendung:

SEIZE name

...

RELEASE name

Speicher

- Modellierung von Mehr-Bedienstationen.
- Einmalige Festlegung der Kapazität bei Simulationsstart.

Verwendung:

```
name STORAGE kap
```

```
...
```

```
ENTER name, anzahl
```

```
...
```

```
LEAVE name, anzahl
```


Aktivatoren

- Entsprechen „quasi“ den Markierungen in den Petrietzen.
- Durchlaufen alle Blöcke des Simulationsmodells.
- Komplexe Parametrisierung möglich.
- Erzeugung durch GENERATE-Block:
GENERATE mw, abw bzw. GENERATE RVvert(...)
- „Bearbeitung“ durch ADVANCE-Block:
ADVANCE mw, abw
- Zerstörung durch TERMINATE-Block:
TERMINATE (-start)

Zufallszahlengenerator

- Lineare Kongruenzmethode (Lehmer-Algorithmus).
- Beliebig viele Zufallszahlenströme möglich.
- Standardmäßig beginnt der n -te Zufallszahlenstrom mit der $100\,000 \cdot n$ -ten Zufallszahl.

$$x_{i+1} = (742\,938\,285 \cdot x_i) \bmod (2^{31} - 1) \quad (1)$$

$$x_0 = 266\,301\,881 \quad (2)$$

Warteschlangen

- „Tracking“ von Aktivatoren, welche sich durch (Abschnitte des) Simulationsmodells bewegen.
- Eintritt in Warteschlange durch QUEUE-Block:
QUEUE name, anzahl
- Austritt aus Warteschlange durch DEPART-Block:
DEPART name, anzahl

Steueranweisungen

Die wesentlichen Steueranweisungen sind

- SIMULATE,
- START und
- END

und dienen zur globalen Steuerung des Simulationsablaufs.

Ungewohnt: Steueranweisungen und Blöcke dürfen im Simulationsmodell beliebig gemischt werden.

Beispiel

zollamt.gps

Agenda

- 1 Einleitung
- 2 Petrietze
- 3 GPSS/H
- 4 Fazit**

Was haben wir gelernt?

- Petrietze sind ein mächtiges Werkzeug zur Darstellung und Simulation paralleler bzw. nebenläufiger Prozesse.
- Petrietze sind anschaulich und formal exakt spezifizierbar.
- GPSS/H kann zur Simulation von Warteschlangenmodellen eingesetzt werden.
- Ähnlichkeiten zu Petrietzen sind vorhanden.
- Ungewohnte Syntax, dafür sehr flexibel.

Fragen?

Vielen Dank für Ihre Aufmerksamkeit.

http://www.rsiegfried.de/downloads/htwm/2006_Petrietze.pdf