Petri-Netze...

Arwed Mett, Dominic Gehrig, Tobias Dorra

15. Juni 2015

Inhaltsverzeichnis

1 Einleitung

Der Formalismus der Petri-Netze ging aus der Theorethischen Informatik hervor und diente ursprünglich lediglich der Beschreibung von vernetzten technischen Systemen.

Meist werden technische Systeme mit Hilfe eines Automaten beschrieben. Mit der ansteigenden Komplexität großer System wächst die Anzahl der Zustände potentiell exponentiell an. Dies wird besonders bei der Modellierung von vernetzen Systemen schnell unübersichtlich. Um trotzdem solche Systeme modellieren zu können, werden sie mit Hilfe von Petri-Netzen beschrieben. Der Formalismus der Petri-Netze beschränkt sich heutzutage nicht mehr nur auf die Themenbereiche der Informatik, sondern hat sich auch als bewährtes Mittel bei der Modellierung von Geschäftsprozessen etabliert. Die große Stärke der Petri-Netze besteht

darin nebenläufige Ereignisse darzustellen, wodurch so genannte „deadlocks“ [1] in Geschäftsprozessen frühzeitig erkannt werden können und die Laufzeit eines Systems durch Analyse verkürzt werden kann. Außerdem lassen sich komplexe Systeme durch eine Modellierung besser verstehen, wodurch z.B. ein Auftraggeber ohne allzu große Details über die Implementierung eines Prozesses diesen nachvollziehen kann. Im Verlaufe dieses Dokumentes soll hauptsächlich auf die Modellierung von Geschäftsprozess eingegangen werden.

Das Dokument erklärt was Petri-Netze im allgemeinen sind und welche Variationen sich im Verlaufe der Geschichte entwickelt haben. Außerdem werden Algorithmen beschrieben, die sich auf Petri-Netze anwenden lassen. Die Theorie und insbesondere die damit verbundenen Algorithmen werden anhand der Planung einer Produktionsanlage für einen Baukastens mit Hilfe von Petri-Netzen modellhaft angewandt.

1.1 Entwicklung

2 Theorie

2.1 Aufbau von Petrinetzen

Im Prinzip können Petri-Netze als Graphen aufgefasst werden. Diese Graphen besitzen allerdings besondere Knoten, die als Komponenten bezeichnet werden. Im folgenden werden die Komponenten eines Petri-Netzes beschrieben.

Plätze sind die passiven Komponenten eines Petri-Netzes. Sie dienen repräsentieren Zustände oder lagern Objekte. Ein Platz wird durch einen Kreis dargestellt:

Leerer Platz Gefüllter Platz

Abbildung 1: Beispiel für einen leeren Platz und einen Platz mit einer (abstrakten) Marke.

Die Objekte, die von Plätzen gelagert werden, werden allgemein als **Marken** bezeichnet. Grundsätzlich kann ein Platz beliebig viele Marken enthalten. Außerdem können in einem Modell unterschiedliche Typen von Marken verwendet werden. Eine Marke kann ein konkretes Objekt aus der realen Welt repräsentieren (Maschinen, Werkstücke, Menschen, ...). Es können jedoch auch abstrakte Marken verwendet werden, die dann meist als schwarzer Punkt dargestellt werden, wie in Abbildung 1 zu sehen. Abstrakte Marken sind sehr nützlich, um Zustände, in dem sich das System befinden kann, zu modellieren.

[1] https://technet.microsoft.com/en-us/library/ms177433(v=sql.105).aspx [Stand: Juni 2015]

Transitionen sind die aktiven Komponenten eines Petri-Netzes. Sie beschreiben eine elementaren Aktion, die Dinge erzeugen, transportieren, verändern oder vernichten [2]. Eine Transition wird durch ein Quadrat dargestellt:

Abbildung 2: Beispiel für eine Transition

Transitionen können zusätzlich noch mit einer Formel beschriftet werden. Diese ist eine Vorbedingung, die erfüllt sein muss, damit die Transition aktiviert werden kann. Wann genau eine Transition aktiviert ist und was das bedeutet, wird später in Kapitel 2.2 noch diskutiert.

Kanten sind selbst keine Komponenten, sondern stellen Beziehungen zwischen Komponenten her. Kanten werden durch einen Pfeil gekennzeichnet und verbinden immer eine Transition mit einem Platz oder einen Platz mit einer Transition. Niemals jedoch verbinden sie zwei Plätze oder zwei Transitionen.

Kanten, die von einer Transition auf eine Stelle zeigen, drücken aus, dass diese Transition Objekte erzeugt und sie in der Stelle ablegt. Kanten, die von einer Stelle auf eine Transition zeigen, drücken aus, dass diese Transition Objekte aus der Stelle entnimmt.

Eine Kante kann zusätzlich noch mit einem Term beschriftet werden. Das Ergebnis dieses Terms müssen Marken sein. Damit wird angegeben, wie viele und welche Marken die Transition aus der Stelle entnimmt beziehungsweise in ihr erzeugt. Im Normalfall ist die Beschriftung einfach ein konstanter Term, es sind jedoch auch Terme mit beliebigen Variablen erlaubt.

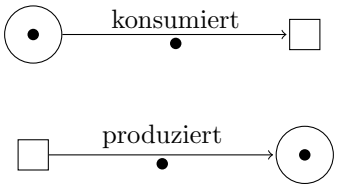


Abbildung 3: Beispiele für Kanten

2.2 Ablaufregeln

Der allgemeine Aufbau von Petrinetzen ist nun bekannt. Nach welchen Regeln sich ein Petrinetz genau verhält, wurde jedoch noch nicht festgelegt. In den folgenden Abschnitten sollen diese Regeln erläutert werden.

Als **Markierung** eines Petrinetzes bezeichnet man die Verteilung von Marken

[2] https://www2.informatik.hu-berlin.de/top/lehre/WS05-06/se_systementwurf/Petrinetze-1.pdf[Stand: Juni 2015]

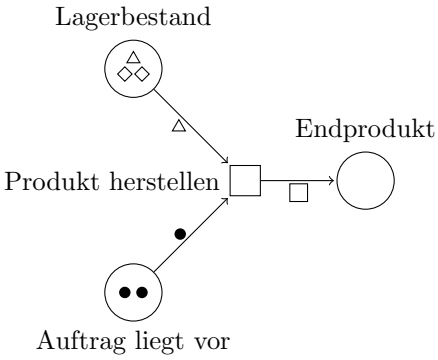


Abbildung 4: Markierung 1 eines Petrinetzes

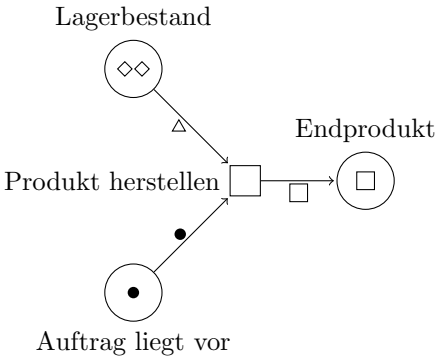


Abbildung 5: Markierung 2 eines Petrinetzes

auf die Plätze. In Abbildung 4 und 5 sind zwei unterschiedliche Markierungen eines Petrinetzes dargestellt.

Als nächstes wird erklärt, wann man eine Transition als **aktiviert** bezeichnet. Anschaulich gesehen ist eine Transition t in einer bestimmten Markierung aktiviert, wenn alle Plätze, aus denen t Marken entnimmt, dafür genug Marken beinhalten. Formal muss also jeder Platz, von dem eine Kante auf t zeigt, von jedem Markentyp genausoviele oder mehr Marken beinhalten als in der Beschriftung der Kante angegeben.

Im Petrinetz aus Abbildung 4 entnimmt die Transition $Produktherstellen$ eine dreieckige Markierung aus der Stelle *Lagerbestand* und eine schwarze Markierung aus $Auftragliegtvor$. Beide Stellen enthalten die geforderten Marken. Also ist die Transition aktiviert.

In Abbildung 5 ist das selbe Petrinetz in einer anderen Markierung dargestellt. Hier enthält $Auftragliegtvor$ noch genug schwarze Marken. Aus dem Platz *Lagerbestand* entnimmt die Transition eine dreieckige Marke, aber in

dem Platz sind nur Rautenfürmige Marken enthalten. Daher ist die Transition in Markierung 2 nicht aktiviert.

Diese Vorgehensweise funktioniert für Kanten, die mit konstanten Termen beschriftet sind. Wenn im Term einer Kante Variablen vorkommen, muss zunächst eine Belegung der Variablen mit echten Werten gefunden werden, für die diese Bedingung zutrifft. Außerdem muss die Formel in der Transition für die Variablenbelegung wahr sein. Wenn eine passende Variablenbelegung existiert, dann ist die Transition in der entsprechenden Markierung unter dieser Variablenbelegung aktiviert. Andernfalls ist die Transition nicht aktiviert.

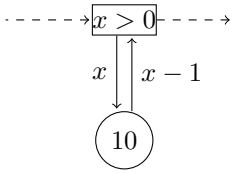
□

Abbildung 6: Ein Petrinetz mit Variablen in den Kantenbeschriftungen

Schnittstellen ermöglichen auf Aktivitäten aus der Umgebung einzugehen. Mit Hilfe der bisherig vorgestellten Bestandteile lassen sich bereits geschlossene Systeme modellieren. Sollen allerdings Interaktionen eines Geschäftsprozesses mit z.B. einem Kunden modelliert werden, benötigt man Schnittstellen. Diese werden durch eine Transition dargestellt.

Kalte Transitionen dienen der Modellierung von Transitionen, bei denen nicht klar ist ob sie eintreffen. Wird z.B. ein Prozess von einem Kunden durch dessen Bezahlung ausgelöst, ist im Vorhinein nicht klar ob dieses Ereignis jemals eintreffen wird. Ist das Gegenteil der Fall wird die Transition als „warm“ bezeichnet. Kalte Transitionen werden durch eine Transition dargestellt, die ein ϵ beinhalten dargestellt.

Zähler erleichtern insbesondere die Modellierung von Lagern. Können z.B. in einem Lager maximal 10 Stücke eines Materials gespeichert werden. Um dies anschaulich zu modellieren wird der Transaktion ein Parameter x übergeben. Dabei wird die Transition als Funktion aufgefasst und mit Rückgabewert $x - 1$.



Somit lässt sich verhindern, dass die Transition mehr als 10 mal ausgeführt wird.

2.3 Variationen

2.4 Algorithmen

2.4.1 Verifikation

2.4.2 Analyse

3 Fallstudie

Im Folgenden soll eine Produktionsanlage eines Baukastens für einen Anemometer [3] modelliert werden, welches im Rahmen eines Projektes [4] der Gruppe Woodchucks an der DHBW Mannheim entwickelt wurde.

3.1 Modellierung

3.2 Verifikation

3.3 Analyse

4 Fazit

Literatur

[3] Ein Gerät zum messen der Windstärke
[4] http://mett.ddns.net [Stand: Juni]

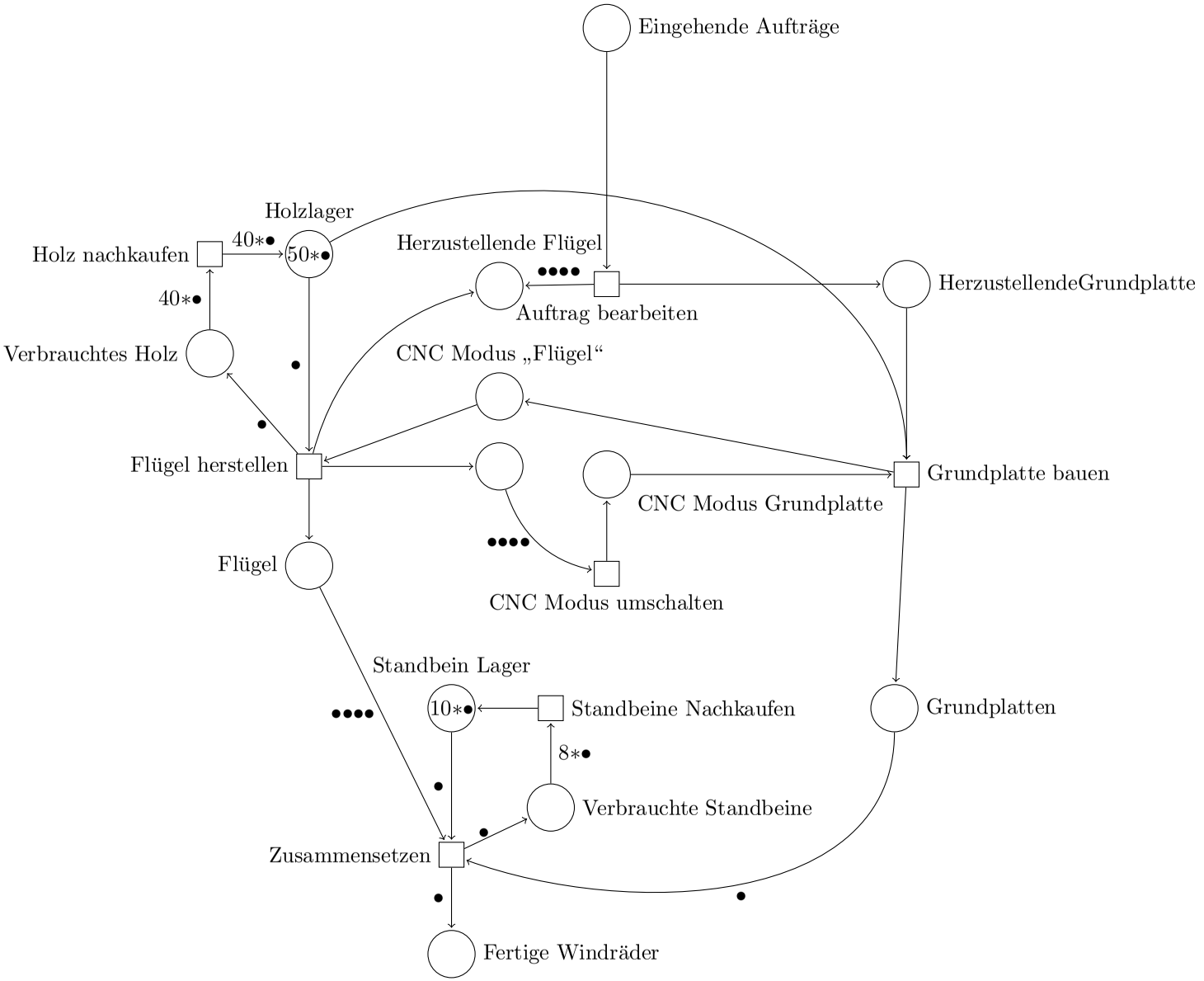


Tabelle 1: Modell der Anemometer Produktion