

## SZENARIEN DER VERTEILTEN UND REAKTIVEN SYSTEME

Wolfgang Reisig  
Humboldt Universität zu Berlin

### Zusammenfassung

Ein verteiltes oder reaktives System hat oft die Aufgabe, Szenarien zu realisieren. Technisch wird ein Szenario als endlicher, verteilter Ablauf konstruiert. Ein beliebiger verteilter Ablauf eines Systems besteht häufig aus mehreren Instanzen einiger seiner Szenarien. Das Verständnis der Szenarien eines Systems ist oft der intuitiv beste Weg zum Verständnis des gesamten Systems. Wir zeigen dies am Beispiel des Modells zum wechselseitigen Ausschluss, des Crosstalk-Algorithmus und des Keksautomaten.

**Schlüsselworte:** Verteilte Systeme. Petri Netze. Szenarien. Modellierung. Crosstalk-Algorithmus.

### 1. Die Idee von Szenarien

Ein Szenario eines Systems ist ein typischer, "gemeinter", aus mehreren Aktionen des Systems zusammengesetzter endlicher Verhaltensstrang, der eine wesentliche Aufgabe des Systems bewältigt. Häufig ist ein Szenario ein Interaktionsmuster zwischen einem Prozess und seiner Umgebung, oder zwischen zwei Prozessen.

In der Regel stimmen Anfangs- und Endzustand eines Szenarios überein (wenn man die Aktualisierung von Daten ignoriert). Ein Ablauf eines verteilten, reaktiven Systems ist oft aus vielen Instanzen einiger weniger endlicher Szenarien komponiert. Wenn man jeden Ablauf so konstruieren kann, ist das System szenarienbasiert.

### 2. Die formale Definition

In der strengsten Form ist ein *Szenario* eines Systemnetzes  $N$  ein endlicher, verteilter Ablauf  $K$  von  $N$ , für den gilt:

- nicht nur der Anfang  ${}^0K$  von  $K$  (die Plätze ohne eingehende Pfeile [1,2]), sondern auch das Ende  $K^0$  (die Plätze ohne ausgehende Pfeile) repräsentieren die Anfangsmarkierung  $M_0$  von  $N$ ;
- in  $K$  wird  $M_0$  nur am Anfang und am Ende erreicht.

Abbildung 1 zeigt die beiden Szenarien des Modells des wechselseitigen Ausschlusses: Eins für den linken und eins für den rechten Prozess. Jedes durchläuft für seinen Prozess den Zyklus von lokal über wartend und kritisch zurück zu lokal.

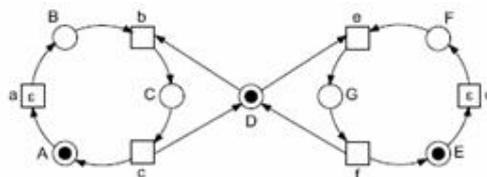


Abb.1-a. Modell des wechselseitigen Ausschlusses

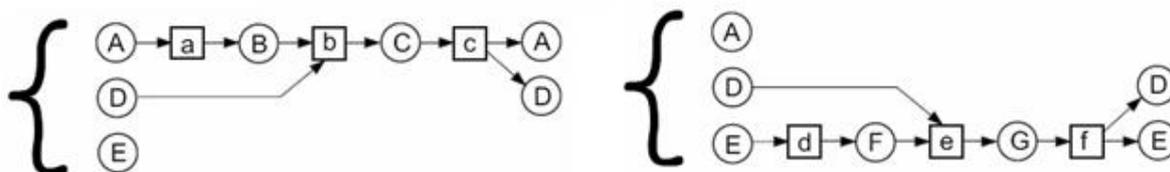


Abb.1-b. Szenarien des Systems des wechselseitigen Ausschlusses

Für verteilte Abläufe  $K$  und  $L$  ist die – intuitiv nahe liegende – Komposition  $K*L$  definiert, wenn das Ende  $K^0$  von  $K$  und der Anfang  ${}^0L$  von  $L$  die gleiche Markierung repräsentieren: Dann identifiziert man gleich beschriftete Plätze von  $K^0$  und  ${}^0L$  miteinander.

$K*L$  enthält alle Elemente von  $K$  und  $L$  und übernimmt deren Ordnung. Die Konstruktion ist eindeutig, wenn  $M$  für jeden Platz  $p$  jedes Element des Universums höchstens einmal enthält, wenn also  $M(p)$  eine "normale" Menge ist. Abb.2 zeigt zwei komponierbare Abläufe  $K$  und  $L$  des Systems des wechselseitigen Ausschlusses.

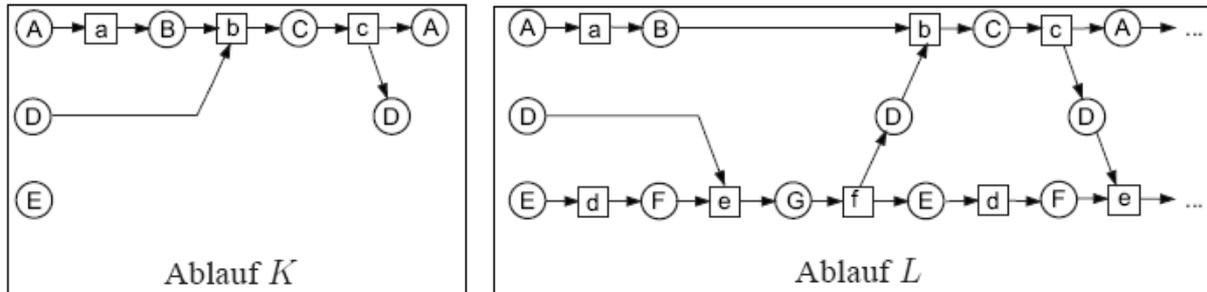


Abb.2. Zwei komponierbare Abläufe

Ihre Komposition  $K*L$  ist der Ablauf in Abb.3.

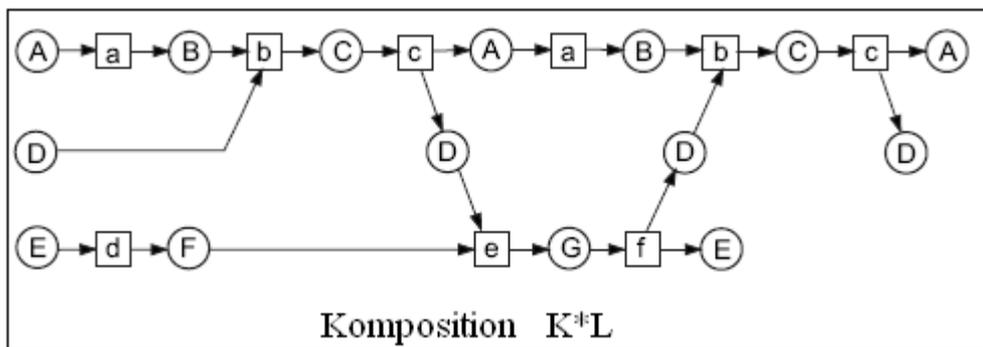


Abb.3

Wer es technisch formuliert sehen möchte, definiert zunächst die *Komponierbarkeit* von  $K$  mit  $L$ . Dazu nutzen wir die offensichtliche Beobachtung aus, dass es auf die Elemente eines Kausalnetzes  $L$  nicht wirklich ankommt und  $L$  durch ein isomorphes Kausalnetz  $L'$  mit entsprechender Beschriftung der Elemente ersetzbar ist.

$K$  ist mit  $L$  komponierbar, wenn man  $L$  isomorph so durch ein isomorphes

$$L' = (P_L, T_L, F_L)$$

ersetzen kann, dass mit

$$K = (P_K, T_K, F_K)$$

Für jedes

$$x \in (P_K \cup T_K) \cap (P_L \cup T_L)$$

gilt:

$$x \in K^0 \cap {}^0L \text{ und } l_K(x) = l_L(x).$$

Das Ende von  $K$  stimmt also mit dem Anfang von  $L'$  überein, ansonsten sind  $K$  und  $L'$  disjunkt. Wenn  $K$  mit  $L$  komponierbar ist, ist  $K*L$  der Ablauf

$$(P_K \cup P_L, T_K \cup T_L, F_K \cup F_L)$$

mit der Beschriftung

$$l(x) = l_K(x) \text{ für } x \in P_K \cup T_K$$

und

$$l(x) = l_L(x) \text{ für } x \in P_L \cup T_L.$$

Aus der Definition der Komposition verteilter Abläufe folgt, dass zwei Abläufe  $A$  und  $B$ , die in  $M_0$  enden, zu einem Ablauf  $A * B$  komponierbar sind, der ebenfalls in  $M_0$  endet.  $A$  und  $B$  können identisch sein. In diesem Fall wird ein Ablauf zwei mal durchlaufen. Damit können beliebig viele (Instanzen von) Szenarien  $K_0, \dots, K_n$  zu einem verteilten Ablauf  $K_0 * \dots * K_n$  komponiert werden.

### 3. Szenarienbasierte Systemnetze

Ein Systemnetz  $N$  ist szenarienbasiert, wenn  $N$  eine endliche Menge  $A$  von Szenarien hat, so dass jeder endliche verteilte Ablauf  $K$  von  $N$  als

$$K = S_1 * \dots * S_n$$

darstellbar ist, mit  $S_1, \dots, S_n \in A$ , und jeder unendliche verteilte Ablauf  $K$  von  $N$  entsprechend als  $K = S_1 * S_2 * \dots$  darstellbar ist, mit  $S_1, S_2, \dots \in A$ .

Beispielsweise ist das System des wechselseitigen Ausschlusses mit den beiden Szenarien aus Abb.1 szenarienbasiert. Später werden wir sehen, dass auch der Crosstalk-Algorithmus und der Keksautomat szenarienbasiert sind.

### 4. Die Szenarien des Crosstalk-Algorithmus

Die beiden Prozesse  $l$  und  $r$  des Crosstalk-Algorithmus (Abb.4) kooperieren in drei Szenarien:

- $l$  sendet,  $r$  empfängt;
- $r$  sendet,  $l$  empfängt;
- beide senden und empfangen, es entsteht also crosstalk.

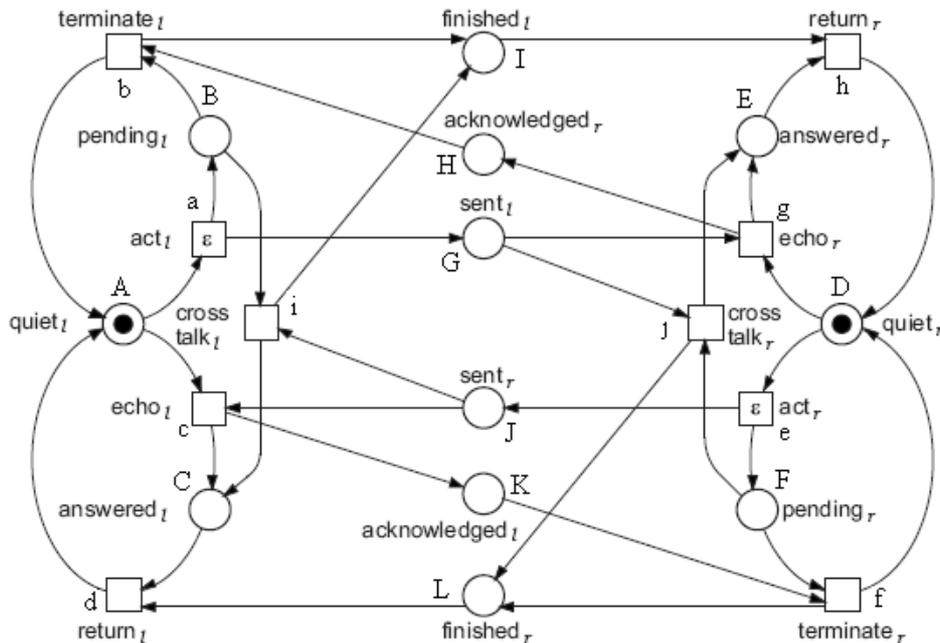


Abb.4. Der Crosstalk-Algorithmus

Man mag eine abstrakte Sicht annehmen und die beiden ersten Szenarien zu einem Szenario "einer sendet, einer empfängt" zusammenfassen. Abb.5 zeigt die entsprechenden verteilten Abläufe. Man sieht leicht, dass der Crosstalk-Algorithmus szenarienbasiert ist, mit den Szenarien.

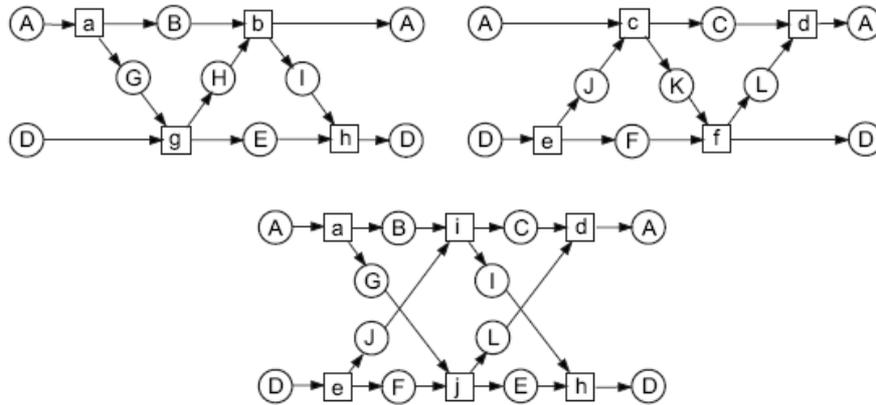


Abb.5. Die drei Szenarien des Crosstalk-Algorithmus

### 5. Die Szenarien des Keksautomaten

Der Keksautomat hat keine Szenarien im strengen Sinn: Die Anfangsmarkierung wird nie wieder erreicht; es werden ja Schachteln aus dem Speicher abgegeben. Dennoch ist das Verkaufen einer Schachtel - intuitiv - ein Szenario. Es beschreibt die "gemeinte" Interaktion des Automaten mit den Nutzer: Nachdem der Nutzer eine Münze eingeworfen hat, gibt der Automat eine Schachtel ab, die der Nutzer dem Entnahmeschacht entnimmt.

Das Szenario hat *Parameter*, die - in Grenzen - verschiedene Ausprägungen der Instanzen erlauben. Alle Ausprägungen haben die gleiche Netzstruktur (bis auf isolierte Plätze zur Komposition von Instanzen), aber durchaus verschiedene Anschriften an den Plätzen.

Im Keksautomat kann damit die Zahl der Münzen in der Kasse, der Wert des Zählers und die Anzahl der Schachteln im Speicher variieren. Für die Fassung des Keksautomaten in Abb.6 zeigt Abb.7-1 eine Ausprägung des Szenarios; Abb.7-2 zeigt eine weitere.

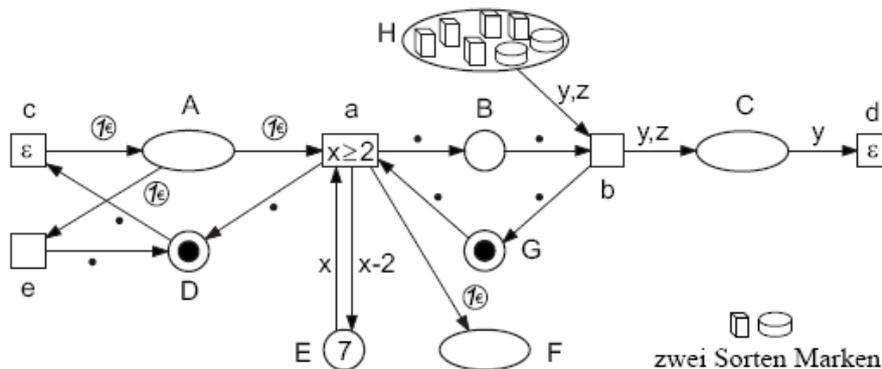


Abb. 6. zwei Sorten Schachteln und Ausgabe zweier Schachteln

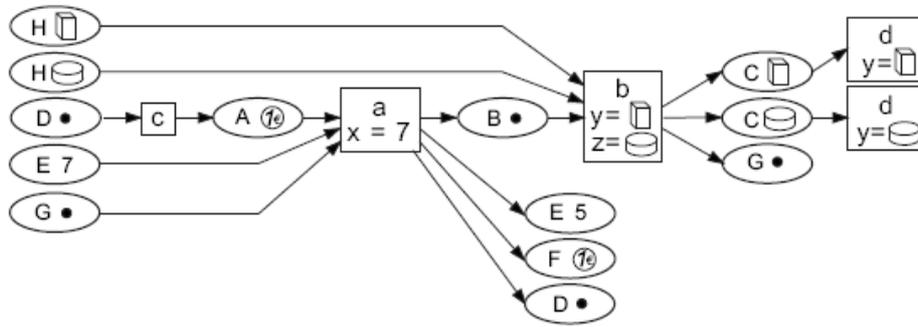


Abb.7-1. Ein verteilter Ablauf des Keksautomaten aus Abb. 6.

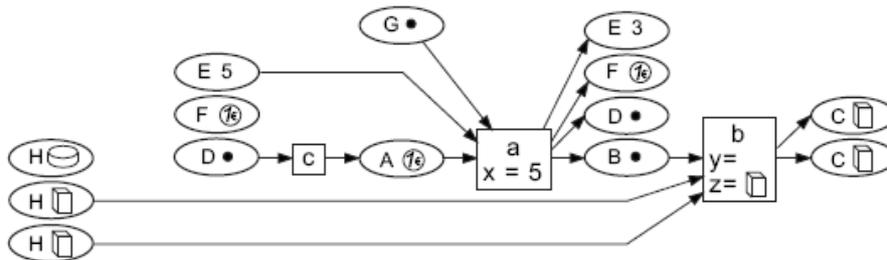


Abb.7-2. Eine weitere Instanz des ersten Szenarios des Keksautomaten

Die Endmarkierung von Abb.7-1 ist mit der Anfangsmarkierung von Abb.7-2 identisch. Deshalb können beide kombiniert werden und es entsteht der verteilte Ablauf in Abb.8.

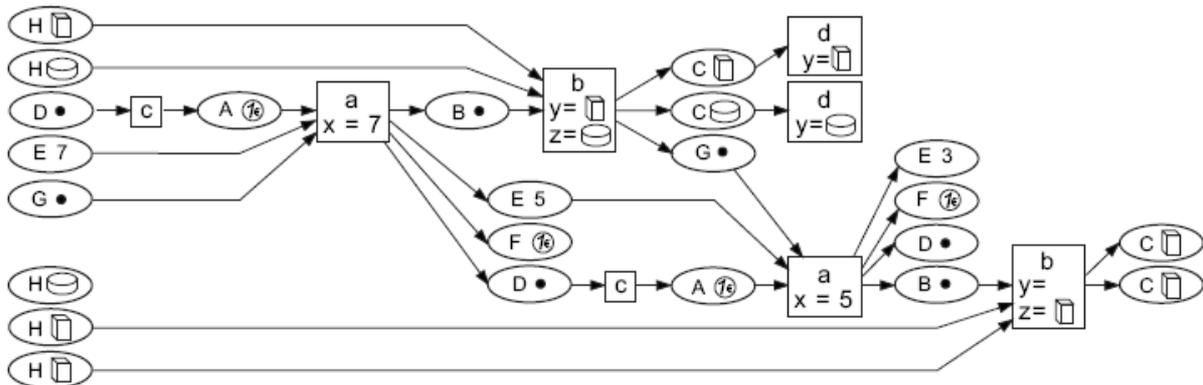
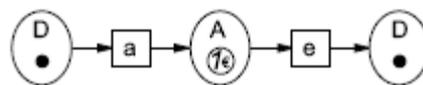


Abb.8. Verteilter Ablauf des Keksautomaten

Neben den verschiedenen Parameterwerten für die Kasse, den Zähler und den Speicher kann eine Ausprägung des Szenarios Aktionen der Transition d enthalten. Das zweite Szenario des Keksautomaten ist sehr einfach:



Auch der Keksautomat ist von Szenarien überdeckt: Jeder verteilte Ablauf ist aus Szenarien kombinierbar.

## 6. Schlussfolgerung

Das Denken in Szenarien kann das Verständnis eines komplexen Systems entscheidend vereinfachen und vertiefen. In der Softwaretechnik beschreibt ein Szenario eine typische Interaktion zwischen einem Nutzer und einem rechnerintegrierten System, die in einem "vernünftigen" Anfangszustand beginnt und mit Erreichen eines (Teil-)Ziels endet. Messages-Sequence-Charts beschreiben Szenarien sehr explizit. Für [3] bilden sie das Fundament einer Modellierungs- und Simulationstechnik.

## Literaturverzeichnis

1. Reisig W. Petri-Netze: Eine neue Einführung. Humboldt Univ. Berlin. 2007
2. Reisig W. Elements of Distributed Algorithms. Springer-Verlag, 1998
3. Desel J., Juhäs G. What is a Petri net ? Informal answers for the informed reader. In: Ehrig, Juhäs, Padberg, Rozenberg (Hrsg.): Unifying Petri Nets. Advances in Petri Nets, LNCS 2128, Springer-Verlag, S. 1-25

## განაწილებული და რეაქტიული სისტემების სცენარები

ვოლფგანგ რეისიგი

ბერლინის ჰუმბოლდტის უნივერსიტეტი

### რეზიუმე

განაწილებული ან რეაქტიული სისტემების დაპროექტების და ანალიზის ამოცანებისათვის ხშირად საჭირო ხდება სცენარების რეალიზაცია. ტექნიკური თვალსაზრისით სცენარის კონსტრუირება წარმოებს სასრული, განაწილებული პროცესების საშუალებით. სისტემის ნებისმიერი განაწილებული პროცესი ხშირად შედგება მისი სცენარის სხვადასხვა ეგზემპლარებისგან. სისტემის სცენარის არსის გაგება ხშირად საუკეთესო საშუალებაა თვით სისტემის აღსაქმელად. ნაშრომში განიხილება ეს საკითხები ურთიერთგამორიცხვის მოდელის, კროსტოლკ-ალგორითმისა და კექსის ავტომატის მაგალითებზე.

## SCENARIOS IN DISTRIBUTED AND REACTIVE SYSTEMS

Reisig Wolfgang

Humboldt University of Berlin

### Summary

Scenarios are widely used tools for designing and analyze of reactive and distributed systems. From technical point of view designing of scenarios must be elaborated by means of finited, distributed processes. Every distributed process of System consists of several scenarios and understanding nature of scenarios is often the best way for understanding whole system. In this paper we consider these items by using examples of Mutual Exclusion, Crosstalk and Cakes Automat.

## СЦЕНАРИИ РАСПРЕДЕЛЕННЫХ И РЕАКТИВНЫХ СИСТЕМ

Рейсиг В.

Берлинский Университет им. Гумбольдта

### Резюме

Для задач проектирования и анализа распределенных и реактивных систем часто требуется реализация сценариев. С технической точки зрения конструирование сценария производится посредством конечных, распределенных процессов. Любой распределенный процесс системы часто состоит из различных экземпляров данного сценария. Понятие сущности сценария системы часто является наилучшим путем понимания самой системы. В работе рассматриваются эти вопросы на примерах взаимоисключающей модели, кросстолк-алгоритма и кекс-автомата.