

## Modellgestützte Analyse und Optimierung Übungsblatt 4

Ausgabe: 27.04.2018, Abgabe: 07.05.2018

### Aufgabe 4.1: Kongruenzgeneratoren

(4 Punkte)

Betrachten Sie den linearen Kongruenzgenerator

$$Z_i = (a \cdot Z_{i-1} + c) \mod m$$

mit  $m = 1000$ ,  $a = 21$ ,  $c = 3$ .

1. Zeigen Sie, dass der Generator volle Periodenlänge hat.
2. Initialisieren Sie den Generator mit  $Z_0 = 871$  und erzeugen Sie 10 Zufallszahlen.
3. Transformieren Sie ihre Zufallszahlen in  $[2,4)$ -gleichverteilte Zufallszahlen.
4. Berechnen Sie den Mittelwert Ihrer "Ziehungen" und vergleichen Sie den Wert mit dem Erwartungswert einer auf  $[2, 4)$ -gleichverteilten Zufallsvariable.
5. Bestimmen Sie die Anzahl der "Runs" in ihrer Zufallszahlenfolge nach dem Runs-Test und vergleichen Sie sie mit dem theoretisch erwarteten Wert für lange Zufallszahlenfolgen.
6. Nutzen Sie ihre Zufallszahlenfolge zur Realisierung von drei exponentialverteilten Zufallszahlen mit Parameter  $\lambda = 0.5$ .

### Aufgabe 4.2: Ziehen von Zufallszahlen aus bestimmten Verteilungen

(4 Punkte)

Formulieren Sie in Pseudocode Routinen, welche Zufallszahlen gemäß folgender Verteilungen erzeugen:

1. Gleichverteilung  $U(a, b)$
2. Exponentialverteilung mit Rate  $\lambda$
3. Geometrische Verteilung mit Parameter  $p$
4. Erlang-Verteilung mit  $k$  Phasen und Rate  $\lambda$
5. dreiphasige Coxverteilung, wobei  $\lambda_i$  der Rate für die Phase  $i$  entspricht. Weiterhin beträgt die Wahrscheinlichkeit für das Durchlaufen der nächsten Phase jeweils 0.5.

Ihnen steht dafür ein Zufallszahlengenerator zur Verfügung, der  $U(0, 1)$  gleichverteilte Zufallszahlen erzeugt.

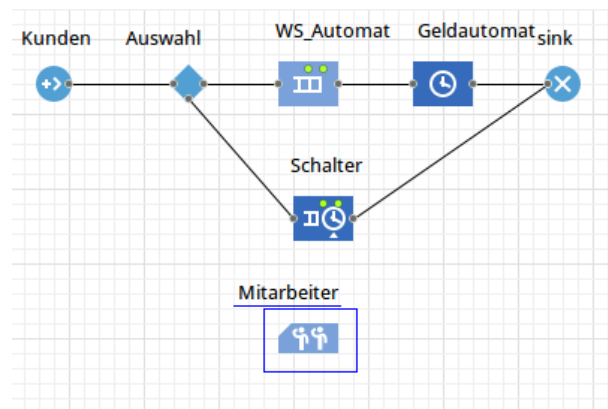
### Aufgabe 4.3: Animation mit AnyLogic

(4 Punkte)

Wir betrachten eine kleine Bankfiliale. Die Filiale besteht aus einem Geldautomaten (für Kunden, die nur Bargeld abheben wollen) und mehreren Schaltern (für Kunden, die Überweisungen und andere aufwendigere Anliegen haben). Es sind folgende Details für die Bank bekannt:

- Kunden kommen exponentialverteilt mit Rate 1.0 pro Minute in der Bank an.
- Mit einer Wahrscheinlichkeit von 50% gehen neue Kunden zum Geldautomaten und mit einer Wahrscheinlichkeit von ebenfalls 50% zu einem Schalter.
- Die Zeit für die Benutzung des Geldautomaten folgt einer Dreiecksverteilung (triangular) mit Parametern 0.8, 1.5 und 3.5 Minuten.
- Es warten maximal 15 Personen gleichzeitig in der Warteschlange des Geldautomaten.
- Um Kunden an den Schaltern zu bedienen, stehen 4 Mitarbeiter zur Verfügung.
- Die Schalter haben eine gemeinsame Warteschlange, in der maximal 20 Personen warten.
- Die Bearbeitungszeit an einem Schalter folgt ebenfalls einer Dreiecksverteilung mit Parametern 2.5, 6 und 11 Minuten.

- a) Modellieren Sie die Bankfiliale mit Hilfe der *Process Modeling Library* in AnyLogic. (Hinweis: Nutzen Sie für den Bankautomaten einen *Delay*-Block und einen *Queue*-Block.) Das fertige Modell könnte folgendermaßen aussehen.



AnyLogic bietet die Möglichkeit, das Modell während der Simulation zu animieren, um den Ablauf anschaulich darzustellen. Die *Process Modeling Library* verfügt über mehrere *Space Markup* Elemente, die hierfür genutzt werden können. Mit diesen Elementen können Punkte, Pfade oder Flächen definiert werden, auf denen sich Ressourcen oder Entitäten (Kunden, Waren) befinden oder über die sie sich bewegen.

- b) Erweitern Sie das Modell der Bankfiliale um eine einfache 2D-Animation. Gehen Sie dazu wie im Folgenden beschrieben vor.
- Zunächst sollen die Kunden visualisiert werden. Wählen Sie dazu die Quelle aus und unter *New Agent* den Punkt *create a custom type*. Wählen Sie die 2D-Animation und dann eine passende Grafik für die Kunden.
  - Verfahren Sie analog mit dem Bankmitarbeiter-Pool (*New resource unit*).
  - Als nächstes nutzen wir die *Space-Markup* Elemente um Bereiche im Modellfenster zu definieren, in denen sich die Kunden und Mitarbeiter aufhalten.

- Platzieren Sie einen *Point Node* und einen *Path* im Modellfenster. Der *Point Node* soll den Geldautomaten repräsentieren, der *Path* die zugehörige Warteschlange. Wählen Sie den *Delay*-Block, der den Geldautomaten darstellt und wählen Sie als *Agent location* den *Point Node*. Verfahren Sie analog für die Queue des Geldautomaten und den Path.
- Platzieren Sie drei *Rectangular Nodes* im Fenster. Diese sollen den Wartebereich für die Kunden, die zum Schalter wollen, den Aufenthaltsbereich für Kunden, die am Schalter bedient werden, und den Bereich der Bankmitarbeiter darstellen. Setzen Sie die *Agent locations* für den Mitarbeiter-Pool und den Service Block entsprechend.
- Wählen Sie in den Eigenschaften der *Rectangular Nodes* für die Mitarbeiter und die Kunden in Bedienung den Punkt *Attractors*. Setzen Sie jeweils 4 *Attractors*. *Attractors* dienen dazu, die Mitarbeiter bzw. Kunden in dem jeweiligen Feld zu positionieren, das heisst die Mitarbeiter/Kunden halten sich jeweils an den *Attractors* auf. In unserem Fall geben die *Attractors* also die Positionen der 4 Bankmitarbeiter und der 4 Kunden an, die gleichzeitig bedient werden können.

c) Simulieren Sie das Modell.