

## Modellgestützte Analyse und Optimierung Übungsblatt 3

Ausgabe: 20.04.2015, Abgabe: 27.04.2015 (12 Uhr)

### Aufgabe 3.1: Simulation eines Supermarkts

(12 Punkte)

An einem Supermarkt treffen “immer wieder” Kunden ein. Jeder Kunde nimmt sich zunächst einen Einkaufskorb, sucht anschließend eine “gewisse Anzahl” von Waren aus, was “eine gewisse Zeit” dauert, wendet sich dann den Kassen zu (wo er u.U. auf Bedienung warten muss) und wird schließlich bedient, was “eine gewisse Zeit” in Anspruch nimmt. Genauere Spezifikation:

- Die Ankunftszeiten der verschiedenen Kunden sind festgelegt durch (zwischen je zwei aufeinander folgenden Ankünften verstreichende) Zwischenankunftszeiten. Die Zwischenankunftszeiten seien beschrieben durch eine exponentialverteilte Zufallsvariable mit Parameter  $\lambda$ .
- Die Zahl der Einkaufskörbe ist begrenzt auf  $M$ . Findet ein eintreffender Kunde keinen freien Korb, dann wartet er bis ein Korb frei wird.
- Die Zahl der Waren die ein Kunde aussucht sei beschrieben durch eine gleichverteilte Zufallsvariable mit Parametern  $[a, b)$ .
- Abhängig von der Anzahl  $k$  auszusuchender Waren benötigt ein Kunde die Zeit  $k \cdot T_s$  für die Warenauswahl und die Zeit  $k \cdot T_p$  für die Bedienung an der Kasse.
- Die Zahl der Kassen ist begrenzt auf  $N$ . Findet ein zahlungswilliger Kunde keine freie Kasse, dann wartet er bis eine Kasse frei wird.
- Fehlen noch Angaben? Treffen Sie in diesem Fall geeignete Annahmen.

Zur Modellierung des Systems sollen die in der Vorlesung vorgestellten Strukturierungsmethoden genutzt werden:

- a) Entwerfen Sie für das oben beschriebene System ein ereignisorientiertes Simulationsmodell nach dem Event-Scheduling Ansatz. Nutzen Sie die Funktionen
- `ziehe_zz(Verteilungstyp, Parameter)` zur Generierung von Zufallszahlen einer gegebenen Verteilung und
  - `plane(Ereignistyp, Ereigniszeit)` zum Einhängen eines Ereignisses in die Ereignisliste.
  - Ferner stehen Ihnen die Funktionen `enqueue(queue, element)` zum Einfügen eines Elements in eine Warteschlange,
  - `first(queue)` zum Auslesen des ersten Elements einer Warteschlange,
  - `dequeue(queue)` zum Löschen des ersten Elements einer Warteschlange und

- `empty(queue)` zum testen, ob eine Warteschlange leer ist, zur Verfügung.

Beschreiben Sie in Ihrer Darstellung, welche Zustandsvariablen benötigt werden, welche Attribute zu einem Kunden gehören, welche Ereignisse auftreten können und wie die Ereignisroutinen aussehen.

- b) Entwerfen Sie für den Supermarkt ein Simulationsmodell nach dem Process-Interaction Ansatz:
- i) aus material-orientierter Sicht (Material sind hier die Kunden)
  - ii) aus maschinen-orientierter Sicht (Maschinen sind hier die Kassen und im weitesten Sinne die Körbe)

Nutzen Sie die Funktionen `pause_for(Zeit)`, um einen Prozess für eine gewisse Zeit zu stoppen, und `wait_until(Bedingung)`, um einen Prozess zu stoppen bis eine Bedingung erfüllt ist. Um einen Prozess (aus der Umwelt) neu zu starten, können Sie die Funktion `start_process(Prozess)` nutzen.