

# Modellgestützte Analyse und Optimierung (SS 2007)

## Übungsblatt 3

### **Aufgabe 3.1:** (zu Kapitel 2.1.2)

(6 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 „schwankend“, beschrieben durch eine Zufallsvariable mit Verteilungstyp  $EXP$  und Parameter  $\lambda$ .
- Die Zahl der Einkaufskörbe ist begrenzt auf  $M$ . Findet ein eintreffender Kunde keinen freien Korb, dann wartet er geduldig bis ein Korb frei wird.
- Die Zahl der Waren die ein Kunde aussucht sei „schwankend“, beschrieben durch Zufallsvariable mit Verteilungstyp  $UNI$  und Parameter  $(a,b)$ .
- 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 geduldig bis eine Kasse frei wird.
- Vielleicht fehlen noch Angaben? Treffen sie in diesem Fall „ihre“ Annahmen.

Entwerfen sie für das oben beschriebene System ein ereignisorientiertes Simulationsmodell nach dem *Event-Scheduling* Ansatz. Nutzen sie die in der Vorlesung dargestellten 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. Machen sie Gebrauch von Pseudo-Code (nach Möglichkeit in Anlehnung an C/C++ oder Java), wann immer ihnen dies geeignet erscheint.

### **Aufgabe 3.2:** (zum Simulationstool „Arena“)

(6 Punkte)

Starten sie Arena und öffnen sie das Beispielmodell „Model 03-01“ aus dem „Book Examples“-Verzeichnis, welches sie vorher in ihr eigenes Verzeichnis kopiert haben (siehe Beschreibung unten).

- a) Lesen sie die Einführung in den Gebrauch von Arena, die sie auf der Webseite der Übungen zur Vorlesung finden. Machen sie sich mit dem Modell vertraut, indem sie die unterschiedlichen Symbole für *Create*, *Process* und *Dispose* im Modell durch Doppelklick öffnen und deren Bedeutung verstehen. Simulieren sie das Modell für die vorgegebenen Einstellungen und betrachten sie die resultierenden Leistungsgrößen. Vergleichen sie die Ergebnisse mit ihren Ergebnissen aus Aufgabe 3.1. Was stellen sie fest?

- b) Simulieren sie das Modell mit 5 Replikationen, indem sie im Menu *Run > Setup > Replication Parameters* entsprechende Anpassungen vornehmen. Betrachten sie den *Category Overview Report* und den *Category by Replication Report*, den sie im Menu *Run > Setup > Reports* wählen können. Erklären sie die beiden Reports und insbesondere den Unterschied zwischen den beiden Reports.
- c) Verlängern sie den Simulationslauf auf 12h. Führen sie keine Replikationen durch. Verändern sie die Ausgabeplots so, dass der Simulationslauf komplett dargestellt wird. Ist ihnen die Darstellung des Simulationslaufs zu langsam? Dann ändern sie die Geschwindigkeit im Menu *Run > Setup > Run Speed*.
- d) Implementieren sie einen neuen Statistik-Kollektor, der die Gesamtzahl an Kunden im System beschreibt (*work in process, WIP*). Informieren sie sich im Arena Hilfe-Menu unter *EntitiesWIP*. Nutzen sie den *Expression Builder* zum einfügen dieser Statistik. Fügen sie dem Modell einen neuen Ausgabeplot für diese Statistik hinzu.

### **Nutzung des Simulationstools „Arena“:**

Zur Installation auf dem heimischen PC kann eine CD mit Arena bei den Übungsgruppenleitern gebrannt werden. Hierzu muss ein entsprechender Rohling mitgebracht werden. Beachten sie dabei die Sprechzeiten.

### **Abgabe der Arena Übungsaufgaben:**

Abgabe der Lösungen zu Aufgabe 2.2 c) und d) per E-Mail an [mao@ls4.cs.uni-dortmund.de](mailto:mao@ls4.cs.uni-dortmund.de)

Das „Betreff“ der E-Mail muss mit dem Text MAO-SS07 + Übungszeit (e.g. 8:00, 10:00 ...) beginnen. Nachfolgend sollte der Name des Abgebenden aufgeführt werden.

Für jeden Aufgabenteil muss eine eigene Datei (Arena Modell .doe-Datei) der E-Mail angehängen werden. Die Dateinamen müssen das Format xxxxxx-32c.doe bzw. xxxxxx-32d.doe, für Aufgabe 3.2 c) bzw. Aufgabe 3.2 d) haben, wobei xxxxxx die 6-stellige Matrikel Nummer des Abgebenden ist.

Bei abgaben für spätere Übungsblätter sind die Dateinamen entsprechend anzupassen.

Nur Abgaben mit korrektem Betreff und korrekten Dateinamen werden bewertet!