

# Modellgestützte Analyse und Optimierung

## Übungsblatt 2

**Ausgabe:** 15. April, **Abgabe:** 24. April, 12 Uhr

### Aufgabe 2.1 (6 Punkte) Simulation

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 unter Umständen 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* mit 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 eine Zufallsvariable mit Verteilungstyp *UNI* und Parameter  $(a, b)$ .
- Abhängig von der Anzahl  $k$  auszusuchender Waren benötigt ein Kunde die Zeit  $t \cdot T_s$  für die Warenauswahl und die Zeit  $t \cdot T_p$  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 und notieren Sie diese tabellarisch.

Entwerfen Sie für das oben beschriebene System ein ereignisorientiertes Simulationsmodell nach dem *Event-Scheduling* Ansatz wie in der Abbildung im Skript auf den Seiten 33 und 34. Machen Sie Gebrauch von Pseudo-Code (nach Möglichkeit in Anlehnung an C/C++ oder Java) und umgangssprachlichen Beschreibungen wie Sie es für sinnvoll halten.

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

Zur Vereinfachung nutzen Sie bitte bei der Verwendung von Warteschlangen folgende Funktionen: Die Funktion `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.

Nutzen Sie die Funktion `ziehe_zz(Verteilungstyp,Parameter)` zur Generierung von Zufallszahlen einer gegebenen Verteilung und die Funktion `plane(Ereignistyp,Ereigniszeit)` zum Einhängen eines Ereignisses in die Ereignisliste.

### **Aufgabe 2.2** (6 Punkte) **zum Simulationstool Arena**

Starten Sie Arena und öffnen Sie das Beispielmmodell „Model 03-01“, welches Sie vorher in ihr eigenes Verzeichnis kopiert haben.

- a) Lesen Sie die Einführung zum 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 *Dispense* im Modell durch Doppelklick öffnen und deren Bedeutung verstehen. Simulieren Sie das Modell für die vorgegebenen Einstellungen und betrachten Sie die resultierenden Leistungsgrößen.
- 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 einer Statistik. Fügen Sie dem Modell einen neuen Ausgabeplot für diese Statistik hinzu.

**Nutzung des Simulationstools Arena:** Für die Nutzung von Arena stehen Ihnen die Rechner an der Fakultät zur Verfügung (z. B. marvin), bei welchen Sie sich mit Ihrem Fakultäts-poolaccount (z. B. nachname00) anmelden können.

Falls Informatik nicht Ihr Hauptfach ist, können Sie bei der IRB einen Pool-Account beantragen. Näheres steht dazu auf den Übungsseiten.

Auf den Rechnern im Fakultätspool starten Sie **Arena** über eine virtuelle Maschine, welche erst dann freigeschaltet werden kann, wenn Sie Ihren Pool-Account der Übungsgruppenleiterin per Email mitteilen. Für die Bearbeitung am heimischen PC stehen CDs mit einer Studentenversion zur Verfügung. Die CD kann dann nach Vereinbarung bei der Übungsgruppenleiterin abgeholt werden.

Wegen Einschränkungen der Studentenversion sind einige Aufgaben nicht durchführbar. Diese Aufgaben können Sie dann nur mit der Vollversion in der VM lösen.

Eine Einführung in den Gebrauch von Arena finden Sie auf der Webseite zur Übung oder im Arena Hilfe-Menu.

**Abgabe der Arena Übungsaufgaben:** Abgabe der Lösungen zu Aufgabe 2.2 c) und d) (*für jede Teilaufgabe jeweils eine eigene Datei*) erfolgt per Email an die Übungsgruppenleiterin.

Vorlesung: [http://ls4-www.cs.tu-dortmund.de/cms/de/lehre/2013\\_ss/mao/index.html](http://ls4-www.cs.tu-dortmund.de/cms/de/lehre/2013_ss/mao/index.html)

Übung: [http://ls4-www.cs.tu-dortmund.de/cms/de/lehre/2013\\_ss/mao\\_uebung/index.html](http://ls4-www.cs.tu-dortmund.de/cms/de/lehre/2013_ss/mao_uebung/index.html)