

Discrete Event System Modelling and Simulation with Matlab/SimEvents

Chris Urbaniak

Master Informatik

Hochschule Bonn-Rhein-Sieg

Grantham-Allee 20, 53757 Sankt Augustin

Email: chris.urbaniak@smail.inf.h-brs.de

Zusammenfassung: Diese Ausarbeitung behandelt die Modellierung und Simulation von Systemen mittels der Matlab-Erweiterung SimEvents.

Zunächst wird in einer kurzen Einführung das SimEvent-Modul gegeben. Anschließend werden einige Begriffe aus dem SimEvents- und Modellierungsumfeld erklärt. Danach erfolgt eine Beschreibung des in SimEvents implementierten Modells, an dem mehrere Experimente vollzogen wurden. Die Ergebnisse werden jeweils präsentiert und ausgewertet. Es folgt ein Einblick in ein Praxisbeispiel indem SimEvents Verwendung findet. Zum Schluss erfolgt ein Fazit.

I. Einführung

SimEvents ist eine Erweiterung des Simulink-Frameworks und erweitert dieses um eine diskrete Event-engine und diskrete Event Komponenten [1]. Bei Simulink handelt es sich um eine Matlab-Erweiterung die es ermöglicht Systeme zu modellieren, mit dem Unterschied zu SimEvents, dass der Fokus nicht auf der diskreten Simulation liegt.

SimEvents sowohl als auch Simulink stammen von den Matlab Entwicklern MathWorks, weshalb eine gute Integration in Matlab gewährleistet wird.

SimEvents ist nicht eigenständig lauffähig, sondern benötigt die Matlab-Umgebung und das Vorhandensein von Simulink. Innerhalb der Matlab-Umgebung ist es möglich auf bestehende Matlab Routinen und Simulink Komponenten zurück zugreifen. Die eigentliche Modellierung mit SimEvents basiert auf dem Drag & Drop Prinzip, bei dem die Komponenten (Blöcke), auf die Modellierungs-oberfläche gezogen werden und mit Pfaden verbunden werden. Bei den Komponenten, die die Erweiterung mit sich

bringt, handelt es sich unter anderem um Warteschlangen, Server, Event- und Entity-Generatoren.

II. Grundlagen

A. Entity

Der Begriff der Entity wird von SimEvents eingeführt, hierbei handelt es sich um Objekte, welche für die Simulation von Interesse sind. Beispiele für Entities sind Kunden in einer Supermarktsimulation, Patienten einer Krankenhaussimulation, oder Lkws einer Logistiksimulation. Zum Zeitpunkt der Modellierung unter SimEvents sind diese Entities nicht sichtbar, da nur Blöcke und deren Pfade angezeigt werden.

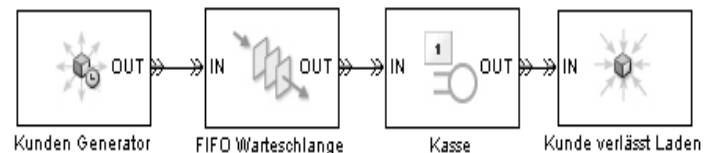


Abbildung 1: Simple SimEvents Anwendung

Wie das Bild zeigt werden keine Entities angezeigt, sondern nur Blöcke und Pfade. Während einer Simulation werden die Entities von einem Block über die vorgegebenen Pfade in den nächsten geschickt, bis sie ihren Verwendungszweck erfüllt haben.

B. Schlüsselkomponenten von SimEvents

Zu den wichtigsten Komponenten zählen die Generatoren, welche Entities erzeugen, die Queues, oder auch Warteschlangen, die die Entities stauen bis die Bedingung zur Weitergabe erfüllt ist und Server, das sind die Blöcke in denen ankommenden Entities bedient werden. In der Realität könnten dies Kassen in einem Supermarkt, Geldautomaten in einer Bank und vieles mehr sein. In SimEvents werden die Bediener auf den Server Begriff reduziert. Das funktioniert weil im Vordergrund nicht die genaue Funktionsweise des Servers steht, sondern, die Zeit der Bearbeitung einer einzelnen Entity, wie viele Entities ein Server gleichzeitig bedienen kann und wohin die Entities als nächstes geschickt werden.

In SimEvents kann zwischen unterschiedlichen Ausführungen der Blöcke gewählt werden. So kann zwischen Generatoren für Entities oder Events, verschiedenen Queues mit unterschiedlichen Strategien; FIFO, LIFO, Priority-Queue, oder aber Single-, n-Server gewählt werden. Ein Single-Server ist in der Lage nur eine Entity auf einmal zu bedienen ein n-Server hingegen n viele, indes n frei gewählt werden darf. Die Eigenschaften der Blöcke kann nach Auswahl über ein Kontextmenü weiter verändert werden.

C. System

„Menge von Komponenten, die in Beziehung stehen und interagieren (z.B. Biotop, Fabrik, Straßennetz mit Fahrzeugen, Sonnensystem mit Planeten)“ [2].

D. Modell & Modellierung

Modelle sind vereinfachte Darstellungen von Systemen, wobei Modelle für die Untersuchung nur die bedeutsamen Komponenten umfassen. Somit ist ein Modell per Definition weniger komplex als sein Ursprungssystem, wodurch es meist erst modelliert (entwickelt) werden kann.

E. Simulation

Simulation bezeichnet das Experimentieren am Modell. Experimente am Modell bergen Vorteile aber auch Nachteile. Zu den Vorteilen zählen:

- Lassen das Ursprungssystem unangetastet und können es daher nicht zerstören.
- Experimente am Modell sind meist kostengünstiger als am Originalsystem.
- Die Parameter sind leichter modifizierbar

Nachteilig ist:

- Es besteht keine Garantie darin, dass Ergebnisse der Simulation auch auf das Originalsystem zurückführbar sind.
- Die Abstraktion auf ein Modell erfordert Wissen über das System, weshalb die Güte eines Modells in Abhängigkeit zum Kenntnisstand des Erstellers steht (beeinflusst den zuvor genannten Punkt).

III. Modellbeschreibung

Für diese Ausarbeitung wurde eigens ein Modell entwickelt und mittels SimEvents implementiert. Dieses Modell und deren Prämissen werden in diesem Abschnitt vorgestellt, auf eine Anleitung zur Implementation wird verzichtet.

Simuliert wird das Besucherverhalten einer geöffneten Hochschulmensa an einem Tag. Die Simulation basiert auf folgendem Modell: Die Mensa hat 3 Stunden geöffnet. Pro Stunde erscheinen 200 Personen (Entities). Neben den verschiedenen Ankunftszeiten, unterscheiden sich Personen in 3 Attributen.

1. Dem gewünschten Gericht. Ein Besucher kann sich zwischen drei Gerichten entscheiden. Burger, Salat und Bockwurst. Die Wahrscheinlichkeit, dass der Besucher sich für Burger entscheidet liegt bei 50%, für Salat bei 30% und schließlich bei 20% für die Bockwurst
2. Der Geduld, die ein Besucher vom Eintreffen bis Erhalt seines Gerichts bereit ist zu warten. Die Geduld liegt gleichverteilt zwischen drei- und fünf-Minuten.
3. Dem Besitz einer Karte. Besucher können bargeldlos per Karte bezahlen. Das Bezahlen mit der Karte geht deutlich schneller von Statten als das Bar bezahlen. Besucher haben zu 40% eine Karte dabei.

Die Attribute eines Besuchers werden bei seinem Erscheinen gesetzt.

Für jedes Gericht gibt es eine eigene Bedienstation. Die Besucher stellen sich je nach gewünschten Gericht an die entsprechende Bedienstation an. Die Schlangen der Bedienstationen basieren auf dem FIFO-Prinzip (First in First out). Die Länge der Schlange ist unbegrenzt. Das Austeilen des Essens an den Bedienstation dauert verschieden lang, so wird davon ausgegangen, dass das Austeilen des Salats (60 Sekunden) länger dauert als das Austeilen des Burgers, der Bockwurst (jeweils 45 Sekunden).

Ein Besucher verlässt die Mensa vorzeitig, wenn er länger auf sein Gericht in einer Warteschlange warten muss, als es durch seine Geduld vorgegeben ist.

Hat ein Besucher sein Gericht erhalten stellt er sich an eine von drei Kassen an. Die Auswahl seiner Kasse geschieht in Abhängigkeit eines Vorhandenseins/nicht Vorhandenseins einer Karte. Besitzt der Besucher eine Karte, stellt er sich an die bargeldlose Kasse an, die ausschließlich Kunden mit einer Karte bedient. Die restlichen Kunden müssen mit einer von zwei Bargeld-Kassen vorlieb nehmen. Aufgrund der Tatsache, dass das Bezahlen an der Kartenkasse erheblich schneller geschieht (10 Sekunden), als an einer Bargeld-Kassen (40 Sekunden), stellen sich Karten-Kunden immer an die Kartenkassen an. Die Kassen-Warteschlangen basieren ebenfalls auf dem FIFO-Prinzip.

Hat ein Kunde sein Gericht von der Bedienstation erhalten, wartet er solange bis er bezahlt hat und verlässt die Mensa nicht vorzeitig. Die Länge der Kassenwarteschlange und die Geduld des Kunden haben also keinen Einfluss mehr auf ein vorzeitiges Verlassen.

Hat ein Besucher bezahlt, wird er von der Simulation nicht weiter erfasst.

Anmerkung: Die meisten der Parameter wurden geschätzt. Konkretere Werte wurde zwar angefragt, lagen aber zum Zeitpunkt des Erstellens der Ausarbeitung noch nicht vor.

IV. Ziel der Simulation

Die Simulation soll Auskunft darüber geben, wie viel Zeit die Kunden im Durchschnitt von ihrer Ankunft bis zum Bezahlvorgang benötigen, wie viele Kunden die Mensa vorzeitig verlassen und wie lange ein Burger/Salat/Bockwurst -Kunde durchschnittlich benötigt bis er seine Bestellung erhält. Außerdem gilt es zu untersuchen welche Auswirkungen eine erhöhte Geduldsrate auf die Verlustrate hat. In der Realität wäre es denkbar, dass die Mensabetreiber in zum Beispiel Bildschirme investieren, von denen vorher ein Erwartungswert bekannt ist, um wie viel sich die Geduld eines wartenden Kunden erhöht.

V. Simulationsvorbereitung

Bevor eine Simulation unter SimEvents gestartet werden kann muss die gewünschte Simulationsdauer angegeben werden. SimEvents gibt bei der Simulationszeit keine Einheiten vor, es obliegt also dem Benutzer wie er die angegebene Zeit deutet. Bei dieser Simulation wurden die Zeiten als Angaben in Sekunden gedeutet. Da die Modell-Mensa angelehnt ist an die Mensa der Hochschule Bonn-Rhein-Sieg, ist sie ebenfalls drei Stunden am Tag geöffnet, was 10800 Sekunden entspricht.

Geschätzt erreichen 200 Besucher pro Stunde die Mensa. Der verwendete Entity-Generator, der die Besucher erzeugt verfügt über einen Perioden Parameter, der entscheidet wann die nächste Entity (der nächste Besucher) erscheint. Damit eben jene 200 Kunden pro Stunde erzeugt werden muss die Periode auf 18 gesetzt werden:

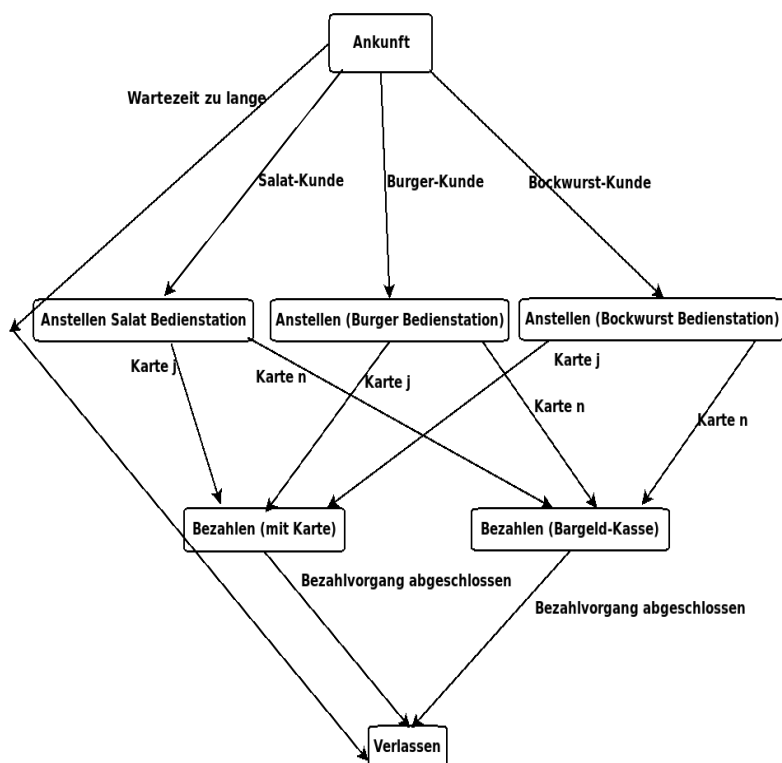


Abbildung 2: Event-Ablaufdiagramm

200 Kunden/Stunde

$$\frac{200}{1}$$

In einer Minute erscheinen 10/3 Besucher

$$\frac{200}{60} = \frac{10}{3}$$

In einer Sekunde erscheinen 1/18 Besucher

$$\frac{10}{3} / 60 = \frac{10}{180} = \frac{1}{18}$$

Daraus folgt: Alle 18 Sekunden erscheint ein Besucher.

VI. Simulations-Durchführung

Start

Die erste Simulation wurde mit folgenden Startwerten gestartet:

Startparameter:

- Simulationsdauer: 10800s (drei Stunden)
- Generations-Periode: 18s (entspricht 200 Kunden pro Stunde)
- P(Burger): 50%
- P(Salat): 30%
- P(Wurst): 20%
- Geduld: 180s-300s
- P(Besitz Karte): 40%
- Bedienzeit Burger: 45s
- Bedienzeit Salat: 60s
- Bedienzeit Wurst: 45s
- Bargeld-Kassen: 2
- Karten-Kassen: 1
- Bedienzeit Bargeld-Kassen: 40s
- Bedienzeit Karten-Kasse: 10s

Ergebnisse:

- Kunden erschienen: 600

Verlorene Kunden:

- Burger-Kunden: 67
- Salat-Kunden: 8
- Wurst-Kunden: 0
- Insgesamt verlorene Kunden: 75

Anzahl bezahlender Kunden:

- Mit Karte: 224
- Ohne Karte bezahlen: 292

- (1) - Burger-Kunden: 237
- Salat-Kunden: 161
- Wurst-Kunden: 118
- Gesamt: 516

Durchschnittliche Wartezeit an der Kasse:

- (2) - Karten-Kunden: 10,5s
- Bargeld-Kunden: 40s

Durchschnittliche Wartezeit für den Empfang des Essens:

- (3) - Aller Kunden: 136,5s
- Wurst-Kunden: 60,5s
- Salat-Kunden: 143s
- Burger-Kunden: 206s

Durchschnittliche Durchlaufzeit:

- Nicht Kartenbesitzer: 228,5s
- Kartenbesitzer: 116,5s
- Wurst-Kunde: 71s
- Salat-Kunde: 163s
- Burger-Kunde: 245,5s
- Alle Kunden: 180s

Auswertung

Es zeigen sich Diskrepanzen bei den durchschnittlichen Durchlaufzeiten, die Ursachen hierfür konnten nicht gefunden. Dennoch zeigen die Durchlaufzeiten eindeutige Tendenzen. Obwohl das Austeilen eines Burgers an einer Bedienstation genauso lange dauert wie das Austeilen einer Bockwurst, verbringen Burger-Kunden im Schnitt 174,5 Sekunden mehr im Mensabedienraum. Ebenso zeigt sich das Verhältnis der Durchlaufzeiten der Karten-/nicht Kartenbesitzer, wobei Kartenbesitzer im Mittel 112 Sekunden eher den Mensabedienraum verlassen als ihr Pendant. Auffällig sind die Zahlen der verlassenden Kunden (75) und der bezahlenden Kunden (516). Addiert ergeben die Zahlen 591 und nicht wie vielleicht zu erwarten gewesen wäre 600. Dieser Umstand ist der Tatsache geschuldet, dass sich zum Ende der Simulation noch Kunden in den Warteschlangen und Servern befinden.

2. Simulationslauf

In diesem Simulationslauf soll nun gezeigt werden, wie sich die Verlustrate verändert, wenn der Mensabetreiber in der Lage ist, die Geduldsrate seiner Besucher anzuheben. Wie bereits erwähnt könnte dieser Bildschirm oder ähnliches installieren. In diesem

hypothetischen Fall erhöht sich die Geduldsrate um eine Minute.
Die weiteren Parameter der Simulation bleiben unangetastet.

Startparameter:

Geduld: 240s-360s

Ergebnis:

- Verlust von Burger-Kunden: 66
- Verlust von Salat-Kunden: 5
- Verlust von Wurst-Kunden: 0
- Gesamt: 71

Auswertung 2. Simulationslauf

Obwohl die Geduld der Kunden um eine Minute erhöht wurde, verringert sich die Anzahl der Kunden die den Mensabedienraum vorzeitig verlassen nur um vier Personen.
Die Maßnahme scheint also wenig erfolgversprechend zu sein.

3. Simulationslauf

Ebenfalls von Interesse könnte für den Mensabetreiber sein, wie sich die Aufenthaltszeiten verändern, wenn dieser eine Kasse schließt und an den verbleibenden beiden Kassen sowohl bargeldloses Bezahlen, als auch bar Bezahlen ermöglicht.

Startparameter:

Hybride Kassen: 2

Geduld: 180s-300s

Ergebnis:

Durchschnittliche Wartezeit:

Kunde ohne Karte: 230s

Kunde mit Karte: 116,5s

Alle Kunden: 181s

Auswertung

Auch hier gibt es wieder eine Ungleichheit zwischen der durchschnittlichen Wartezeit aller Kunden und der der Karten-/ nicht Kartenbesitzer. Das eigentlich Interessante ist jedoch, dass sich die Zeiten im Vergleich zu den drei Kassenmodell nur marginal geändert haben. Zum Vergleich im ersten Simulationsdurchlauf benötigten die Kunden ohne Karte im Schnitt 228,5 Sekunden und in diesem Versuch 230 Sekunden. Die Werte der Kartenbesitzer hat sich

nicht verändert und liegt bei beiden Durchläufen bei 116,5 Sekunden.

Hier kann also dem Mensabetreiber eine klare Empfehlung ausgesprochen werden. Zumal die leicht erhöhten Aufenthaltszeiten keinerlei Auswirkungen auf das frühzeitige Verlassen der Kunden haben, weil die Annahmen getroffen wurden, dass sobald ein Kunde sein Essen erhalten hat, er die Mensa nicht mehr frühzeitig verlässt und die Warteschlangen keine Begrenzung in ihrer Länge aufweisen.

VII. Simulationsfazit

Der dritte Simulationsdurchlauf ist eine Demonstration der Bedeutung von Simulationen. Es ist nicht intuitiv, dass das Schließen einer Kasse so geringe Auswirkungen auf die Besuchsdauer hat. Auch könnte durchaus gefolgert werden, dass sich die Aufenthaltszeit der Kartenbesitzer durch das Teilen der Kassen mit den anderen Besuchern als große Entbehrung erweisen würde, was jedoch nicht der Fall ist. In der Realität ist so ein Experiment ohne Risiko kaum durchführbar und könnte mit herben Verlusten enden, weshalb ein Modell meist vorgezogen wird. Dennoch darf nicht unerwähnt bleiben, dass es sich gerade bei diesem Modell um eine starke Abstraktion eines Originalsystems handelt, inwieweit die Erkenntnisse Rückschlüsse auf das Ursprungssystem erlauben ist fraglich. Im Vordergrund dieser Seminausarbeitung stand allerdings nicht die Erstellung eines möglich realitätsnahem Modells, sondern unter anderem die Frage inwieweit SimEvents als Modellierungs- Simulationsmodul geeignet ist, worauf in den nächsten Abschnitten eingegangen wird.

VII SimEvents Praxisbeispiel

Marius Gemeinhardt beschreibt in seinem Artikel „Optimizing Automotive Manufacturing Processes with Discrete-Event Simulation“ [4], wie er mithilfe von SimEvents den Fertigungsprozess von dem Unternehmen Daimler optimieren konnte. Dafür mussten

Arbeiter, Maschinen und Fahrzeuge richtig koordiniert werden, eine Aufgabe die bei den Konkurrenten meist durch erfahrene Ingenieure übernommen wird.

Die vorhandene Komplexität macht es den Ingenieuren allerdings unmöglich sämtliche Ausmaße von tendenziellen Optimierungen zu erfassen, weshalb sich eine Simulation anbietet. Die Simulationsplattform von Gemeinhardt auf der Simulink und SimEvents laufen, wird mit riesigen Datenmengen von Teststationen gespeist, an denen ein Auto vor dem Verlassen der Fertigungshalle vorbei muss. Die Daten werden dann auf der Plattform ausgewertet und die Ergebnisse unterstützend zur Entscheidungsfindung und Prozessoptimierung eingesetzt. Inwieweit die Effizienz gesteigert wurde bleibt der Autor dem Leser schuldig. Interessant ist jedoch, dass renommierte Unternehmen auf SimEvents zurückgreifen, sowie die Tatsache, dass SimEvents die gewaltigen Datenmassen bewältigen und zur Simulation nutzen kann, denn an jeder Teststation werden in etwa 200 000 Zeilen an Information (laut Autor) pro Auto gesammelt und zur Auswertung weitergeleitet.

An enormer Mächtigkeit gewinnt SimEvents durch die Benutzung von Matlab Routinen. Des Weiteren können unter Zuhilfenahme der Simulink Komponenten hybride Modelle erstellt werden. Zu Matlab und dessen Erweiterungen bleibt noch zu sagen, dass diese Produkte nicht frei verfügbar sind, sondern erworben werden müssen. Eine Privatperson müsste um SimEvents nutzen zu können 7850€ (Matlab 2000€, Simulink 3000€ und SimEvents 2850€ Stand 02.07.2017)bezahlen. Ein Student hingegen 76€ („Matlab and Simulink student suite“ 69€, SimEvents 7€)[3].

Die Preise werden hier ohne jegliche Wertung genannt, sollten aber vor dem Erwerb Beachtung finden.

Anmerkung

Für die Erstellung der Simulation wurde SimEvents/Simulink in der R10b 7.6 Version genutzt.

Die genutzte Matlab Version lautet: 7.11.0.584. Als Betriebssystem wurde Microsoft XP Professional mit Service Pack 3 und einer 32 Bit Architektur verwendet.

VIII SimEvents Fazit

SimEvents bietet einen raschen Einstieg in die Modellierung und ist einfach zu bedienen, wodurch gerade recht simple Modelle in kürzester Zeit umgesetzt werden können. Vorwissen ist nur im geringen Maße notwendig, weil die Blöcke ausreichend kommentiert sind und einige Demosimulationen mitgeliefert werden.

Dadurch, dass die Modellierung graphisch geschieht ist sie vermutlich für Außenstehende einfacher zu verstehen, als es bei programmierten Lösungen der Fall wäre. Der Vorteil der Visualisierung schwindet aber mit zunehmender Komplexität der Modellierung, zwar ist das bei der Programmierung nicht anders, dennoch geht der ursprüngliche Vorteil verloren.

Quellen

[1] Kein Autor angegeben.
„Matlab&SimEvents“ (SimEvents Getting Started Guide) September 2015

[2] Norbert Th. Müller
„Einführung in die ereignisgesteuerte Simulation“
Seite 3. Systeme und Modelle
Vorlesung im WS 2002/2003
Skript, Stand: 25. Oktober 2002
Abteilung Informatik Universität Trier

[3] Matlab-Produktpreise. Letzter Zugriff am 02.07.2017 <https://de.mathworks.com/pricing-licensing.html?prodcode=ML&intendeduse=undefined>

[4]Marius Gemeinhardt „Optimizing Automotive Manufacturing Processes with Discrete-Event Simulation“

Letzter Zugriff am 02.07.2017,

2017 veröffentlicht,

<https://de.mathworks.com/company/newsletters/articles/optimizing-automotive-manufacturing-processes-with-discrete-event-simulation.html>