

Einführung in die Simulation mit Arena 5.0

Programmierpraktikum

WS 2002/3

Einführung in die Simulation

- Grundidee:
 - Entwurf eines experimentellen Modells, das sich wie ein zu untersuchendes reales System verhält
 - Durchführung von Versuchen bzw. (Hoch-) Rechnungen im Modell
- Ziel:
 - Sammeln von Informationen über das Verhalten des Systems in verschiedenen Situationen
 - Beantwortung der „Was passiert, wenn ...“-Frage, um mögliche Handlungsalternativen zu finden
 - Analyse des (zukünftigen) Systemverhaltens

Einführung in die Simulation

- Vorteile:
 - Testen der Wirkung von Entscheidungen
 - Laufzeit ist wesentlich kürzer als simulierte Zeit
 - Rekonstruktion von bestimmten Phänomenen durch verschiedene Variablenbelegungen
- Nachteile:
 - Simulationsergebnisse werden häufig mit Zufallsvariablen berechnet => Ausgabeanalyse notwendig
 - Unterschiedliche Ergebnisse durch unterschiedliche Simulationsmodelle => Simulatoren (z.B. Arena)
 - Simulationen sind oft sehr rechenaufwändig

Einführung in die Simulation

- Anwendungsgebiete:
 - Entwurf und Analyse von Herstellungssystemen
 - Ermitteln der Hard- und Softwareanforderungen für ein Computersystem
 - Analyse von Finanz- und Wirtschaftssystemen
 - Bestimmung einer kostengünstigen Auftragspolitik für den Lagerbestand eines Betriebes
 - Entwurf von neuen Kommunikationssystemen
 - Entwurf und Analyse von Transportsystemen
 - Ermittlung des optimalen Aufbaus von Service-systemen (z.B. Krankenhäuser, Banken, ...)

Einführung in die Simulation

- Beispiel Bankschalter:
 - Untersuchung der Serviceleistungen einer Bank
 - Wie viele Schalter sind notwendig, um einen guten Kundenservice zu leisten?
 - Entsprechendes System besteht aus den Schaltern, den Angestellten und den Kunden
 - Attribute jedes Angestellten: „frei“ oder „beschäftigt“
 - Attribute jedes Kunden: Ankunftszeit in der Bank
 - Angestellte sind Ressourcen
 - Kunden sind dynamische Objekte => offenes System

Einführung in die Simulation

- Zustand des System:
 - Zahl der beschäftigten Angestellten
 - Zahl der Kunden in der Bank
 - Ankunftszeit jedes Kunden
- Gesucht ist z.B.
 - die durchschnittliche Wartezeit eines Kunden in der Schlange vor dem Schalter
 - die durchschnittliche Anzahl von wartenden Kunden in der Schlange
 - die durchschnittliche Auslastung des Schalters

Simulation mit Arena

Arena ist ein mächtiges Simulationsprogramm und bietet im Vergleich zu anderer Simulationssoftware eine animierte Allround-Simulation.

Es geht nach dem prozessorientierten Ansatz vor, d.h. Komponenten führen Aktionen durch, wodurch sich das System verändert. Andere Programme simulieren ereignisorientiert, d.h. Ereignisse verändern das System.

Wir wollen mit Arena ein Simulationsmodell erstellen und damit Experimente durchführen.

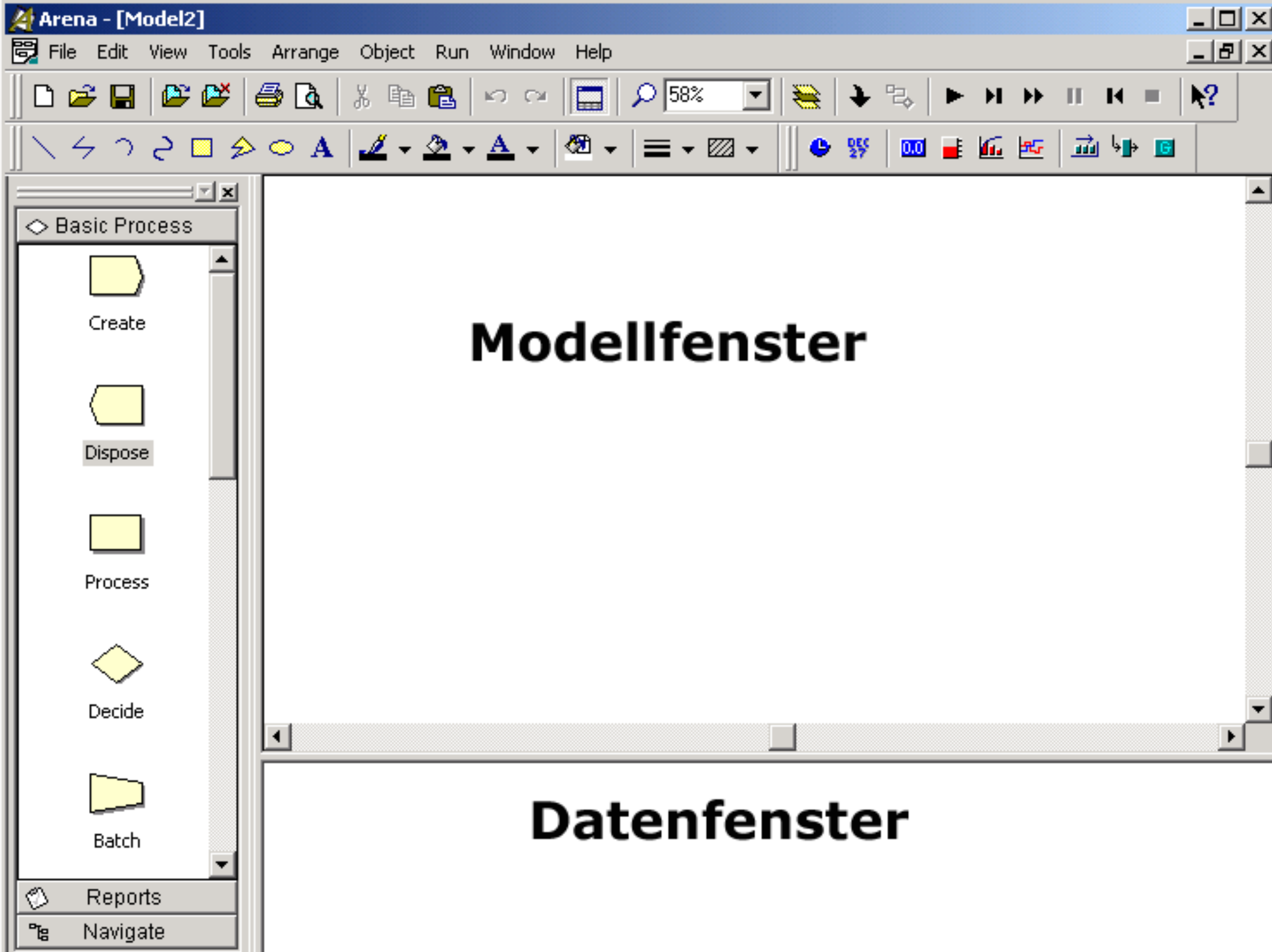
Die folgenden 5 Schritte müssen mit dem Arena-Hauptprogramm ausgeführt werden:

Simulation mit Arena

- Aufbau eines Grundmodells, das aus *Arena modules* besteht, per „drag & drop“ (z.B. eine Warteschlange mit einem Bediener).
- Einstellen und hinzufügen von Parametern und Daten (z.B. Bedienzeiten) durch Doppelklicken auf die Module.
- Simulieren des Modells und Überprüfung, wie gut das Modell das reale System modelliert.
- Analyse der Simulationsergebnisse.
- Änderung des Modells bzw. der Parameter, um mehr über das mögliche Verhalten des realen Systems herauszufinden.

Das Grundmodell erstellen

- Wir benutzen die Formen (Module) aus dem *Basic Process panel*.
- 2 Typen von Modulen: *Flowchart modules* und *Data modules*.
- Flußdiagramm-Module werden in das Modellfenster gezogen und miteinander verbunden.
- Das Flußdiagramm beschreibt dann die Logik des Prozesses.
- Das Modellfenster besteht aus 2 Bereichen: der obere Teil zeigt die Grafik des Modells , der untere die Daten des Modells.



Beispielmodell

- (1) Per drag & drop ein *Create module* aus dem *Basic Process panel* ins Modellfenster ziehen. Es soll einen Ankunftsprozess modellieren (z.B. ankommende Kunden in einer Bank).
- (2) Im unteren Fenster die Datenfelder ändern (*Name*: „Ankunft“, *Entity Type*: „Kunden“, *Type*: „*Random(Expo)*“ mit *Value*: „0.5“). Damit kommen die Kunden mit einer exponentiell verteilten Zwischenankunftszeit an. Im Mittel liegt eine halbe Stunde zwischen zwei Ankünften.
- (3) Per drag & drop ein *Process module* aus dem *Basic Process panel* ins Modellfenster ziehen und rechts neben dem *Create module* platzieren. Es modelliert einen Schalter mit einem Angestellten der Bank, von dem die Kunden bedient werden wollen.

Ankunftsprozess

Arena - [Model3]

File Edit View Tools Arrange Object Run Window Help

58%

Create 1

Create

Dispose

Advanced Transfer

Advanced Process

Reports

Navigate

Create - Basic Process

	Name	Entity Type	Type	Value	Units	Entities per Arrival	Max Arrivals	First Creation
1	Create 1	Entity 1	Random (Expo)	1	Hours	1	Infinite	0.0

Entity 1

(724, 1746)

For Help, press F1

The screenshot displays the Arena simulation software interface. The main workspace shows a yellow 'Create 1' process block. A black arrow points from the 'Create' block in the 'Basic Process' library to this block. The 'Basic Process' library on the left includes 'Create', 'Dispose', and other process blocks. The 'Create - Basic Process' configuration table is visible at the bottom, showing settings for 'Create 1' with a random exponential distribution, a value of 1 hour, and infinite max arrivals. The status bar at the bottom indicates 'Entity 1' and coordinates '(724, 1746)'.

Beispielmodell

- (4) Darauf achten, daß die beiden Module mit einer Linie verbunden sind (, normalerweise durch *Auto-Connect* automatisch der Fall). Sind die Module noch nicht verbunden, muß der *Connect*-Button geklickt werden. Damit wird dann das rechte Ende des *Create modules* mit dem linken Ende des *Process modules* durch Klicken verbunden.
- (5) Das *Process module* mit einem Klick aktivieren und unten im Datenfenster die Felder *Name* auf „Bedienung“ setzen und das Feld *Action* ändern in „Seize Delay Release“. Damit müssen ankommende Kunden warten (*Seize*), werden bedient (*Delay*) und verlassen das System dann wieder (*Release*).
- (6) Das Feld *0 rows* unter *Resources* anklicken im neu aufgehenden Fenster *Resources* per Doppelklick eine neue Zeile erzeugen. Wir setzen *Resource Name* auf „Bediener“, da wir nur einen Schalter haben.

Bedienprozess

Arena - [Model1]

File Edit View Tools Arrange Object Run Window Help

100%

Basic Process

- Create
- Dispose
- Process
- Reports
- Navigate

Ankunft

Bediener

Resources

Quantity

Double-click here to add a new row.

Process - Basic Process

	Name	Type	Action	Priority
1	Bediener	Standard	Seize Delay Release	Medium(2)

0 rows

Triangular	Hours	Value Added			
			5	1	1.5

For Help, press F1

(409, 1261)

The screenshot displays the Arena simulation software interface. The main workspace shows a process flow diagram with two yellow rectangular blocks: 'Ankunft' (Arrival) on the left and 'Bediener' (Server) on the right, connected by a horizontal line. The 'Ankunft' block has a small blue '0' at its bottom right corner, and the 'Bediener' block has a small red '0' at its bottom center. A black arrow points from the 'Resources' table towards the 'Bediener' block. The left sidebar contains a 'Basic Process' panel with icons for 'Create', 'Dispose', 'Process', 'Reports', and 'Navigate'. The bottom of the interface features a status bar with the text 'For Help, press F1' and a coordinate display '(409, 1261)'.

Bedienprozess

The screenshot shows the Arena simulation software interface. The main workspace displays a process flow starting with an 'Ankunft' (Arrival) block, followed by a 'Bedienung' (Service) block. The 'Bedienung' block is currently selected, and a 'Resources' dialog box is open, showing a table with one resource named 'Bediener' with a quantity of 1.

Resources Dialog Box:

	Type	Resource Name	Quantity
1	Resource	Bediener	1

Double-click here to add a new row.

Process - Basic Process Table:

	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Expression
1	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	EXPO(0.1)

Beispielmodell

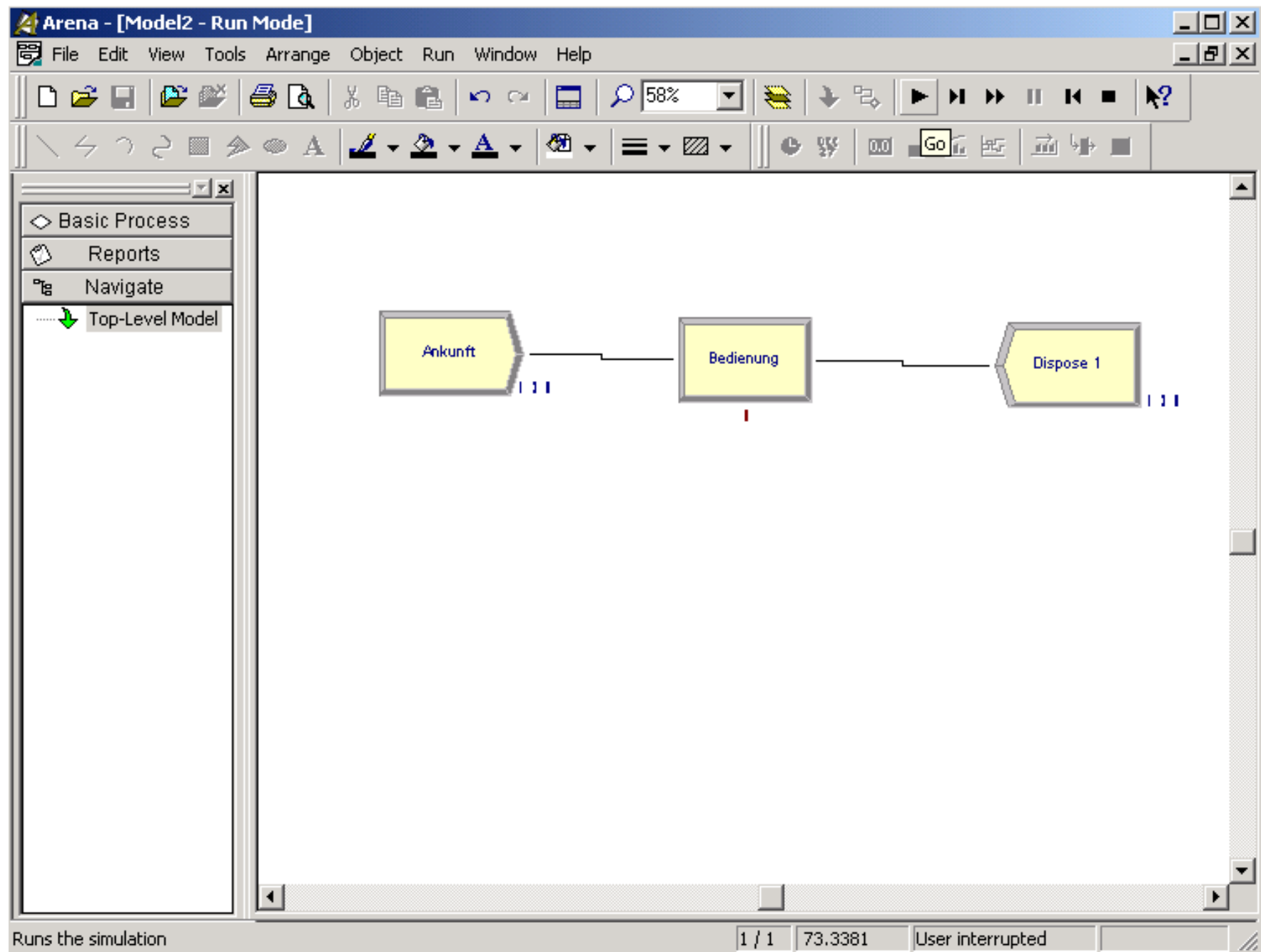
- (7) *Delay Type* setzen wir auf „Expression“, um im Feld *Expression* eine spezielle Wahrscheinlichkeitsverteilung anzugeben. Sie bestimmt, wie lange ein Kunde bedient wird.
- (8) Mit einem Klick auf *Expression* erscheint eine Liste von Verteilungen. Wir wählen „EXPO(Mean)“, um wieder eine exponentiell verteilte Bedienzeit zu erhalten. *Mean* ersetzen wir durch „0.1“. Damit wird ein Kunde im Mittel 6 Minuten bedient.

!Achtung! Die mittlere Bedienzeit sollte bei einer unendlich langen Simulation kürzer sein als die mittlere Zwischenankunftszeit, sonst ist der Warteraum schnell überfüllt.

Beispielmodell

- (9) Nun per drag & drop ein *Dispose module* aus dem *Basic Process panel* ins Modellfenster ziehen und rechts neben dem *Process module* platzieren. Es modelliert den Abgangsprozess der Kunden. Darauf achten, daß das Modul mit dem *Process module* verbunden ist(, ansonsten mit *Connect* nachholen).
- (10) Mit dem *Play*-Button kann die Simulation gestartet werden. Da wir eine keine Stopbedingung für die Simulation eingegeben haben, läuft die Simulation bis der *Pause*-Button geklickt wird.
- (11) Die Zahl unter dem Create(*Dispose*) module sagt uns, wie viele Kunden insgesamt in die Bank gekommen sind(die Bank verlassen haben). Die Zahl unter dem *Process module* zeigt die Anzahl der wartenden Kunden.

Simulation starten



Terminierungsbedingungen

- Über das *Create module* (feste Kundenzahl):

Das *Create module* markieren und im Datenfeld *Max Arrivals* den Wert *Infinite* durch eine feste Zahl ersetzen.

- Über *Run Setup* (feste Zeit):

Im Menü *Run* das *Setup* öffnen und unter *Replication Parameters* die *Replication Length* auf eine feste Zahl setzen.

- Über die *Terminating Condition*:

Ebenfalls im *Run Setup* wird als *Terminating Condition* ein Ausdruck eingegeben und die Simulation stoppt, wenn er erfüllt ist.

Simulieren des Modells

- Oft ist eine sog. Einschwingphase (*Warm-Up Period*) nötig. Das ist der Zeitabschnitt von Beginn der Simulation bis zu dem Punkt, an dem das System alle „Startunregelmäßigkeiten“ abgelegt hat.
- Fängt man z.B. mit einem leeren System an, was aber in der Realität eher selten vorkommt, so muß sich erst ein Normalzustand „einpendeln“.
- Stellt man im Run Setup unter Warm-Up Period die Zeitspanne des „Einpendelns“ ein, so werden erst danach Ergebnisse aufgezeichnet. Die verfälschten Werte fließen nicht in die Endergebnisse ein.
- Startet man die Simulation mit dem *Play Button*, so stoppt sie, wenn die Terminierungsbedingung erfüllt ist oder läuft (theoretisch) unendlich lange.

Arena - [Model2]

File Edit View Tools Arrange Object Run Window Help

58%

Basic Process

- Create
- Dispose
- Process
- Decide
- Batch
- Reports
- Navigate

Diagram:

```

    graph LR
      Ankunft[Ankunft] --> Bedienung[Bedienung]
      Bedienung --> Abgang[Abgang]
  
```

Run Setup

Speed | Run Control | Reports

Project Parameters

Number of Replications: 1

Warm-up Period: 0.0

Replication Length: Infinite

Hours Per Day: 24

Terminating Condition:

Replication Parameters

Initialize Between Replications

☒ Statistics ☒ System

Time Units: Hours

Time Units: Hours

Base Time Units: Hours

OK Abbrechen Übernehmen Hilfe

Process - Basic Process

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Expression	Report Statistics
1	Bedienung	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	EXPO(0.2)	<input checked="" type="checkbox"/>

For Help, press F1

(262, 1725)

Simulationsreport

- Terminiert die Simulation, kommt man (nach einer Bestätigung) sofort zum Simulationsreport.
- Hier werden die Ergebnisse, die während der Simulation gesammelt wurden, angezeigt.
- Im linken Teilfenster steht statt des *Basic Process panel* der *Reports panel*. Dort läßt sich für jeden Modellbestandteil per Doppelklick ein eigenes Reportfenster öffnen.
- Öffnet man den Verzeichnisbaum des Projekts und wählt man *Queue* aus, erhält man unter *Waiting Time* die durchschnittliche Wartezeit eines Kunden in dieser Simulation.
- *Number Waiting* gibt die mittlere Anzahl von Kunden in der Schlange an.

Simulationsreport

- Mehrere Replikationen (Simulationsdurchläufe mit dem selben Modell und identischen Parametern) :
 - Im Run Setup kann man unter *Replication Parameters* den Wert *Number of Replications* > 1 setzen.
 - Simuliert man erhält man im Report Ergebnisse, die sich auf alle Replikationen beziehen und auch Ergebnisse jedes einzelnen Laufs.

!Achtung! Mehr als nur ein Lauf ist in jedem Fall sinnvoll, da die Werte eines einzelnen Laufs meist nicht aussagekräftig genug sind.

Simulationsreport

- In unserem Beispielmmodell sind die exakten Erwartungswerte bekannt:

Die mittlere Anzahl von Kunden in der Schlange ist 0.05 und die durchschnittliche Wartezeit eines Kunden ist 0.025.

- Bei einer Simulation mit z.B. nur einem Lauf und 200 ankommenden Kunden, bekommt man Ergebnisse, die den wahren Wert stark verfehlen (hier 0.0602 als mittlere Anzahl von Kunden in der Warteschlange und 0.0316 als durchschnittliche Wartezeit).
- Simuliert man dagegen mit z.B. 5 Läufen und 500 ankommenden Kunden, erhält man 0.0542 als mittlere Anzahl von Kunden und 0.0263 als durchschnittliche Wartezeit.

Preview

[-] Beispielprojekt

+ Entity

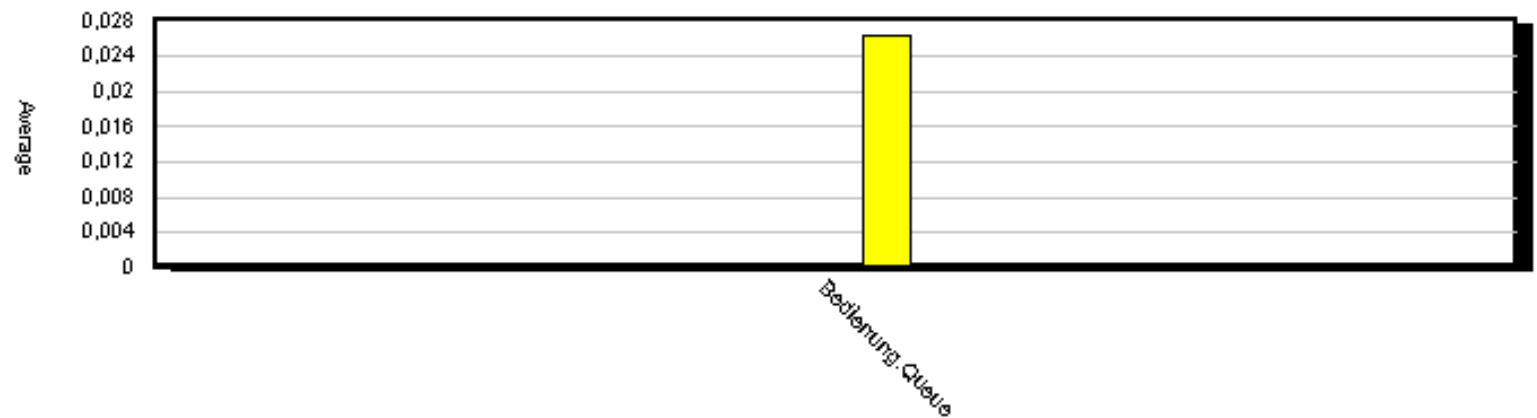
+ Queue

+ Resource

Time

Waiting Time

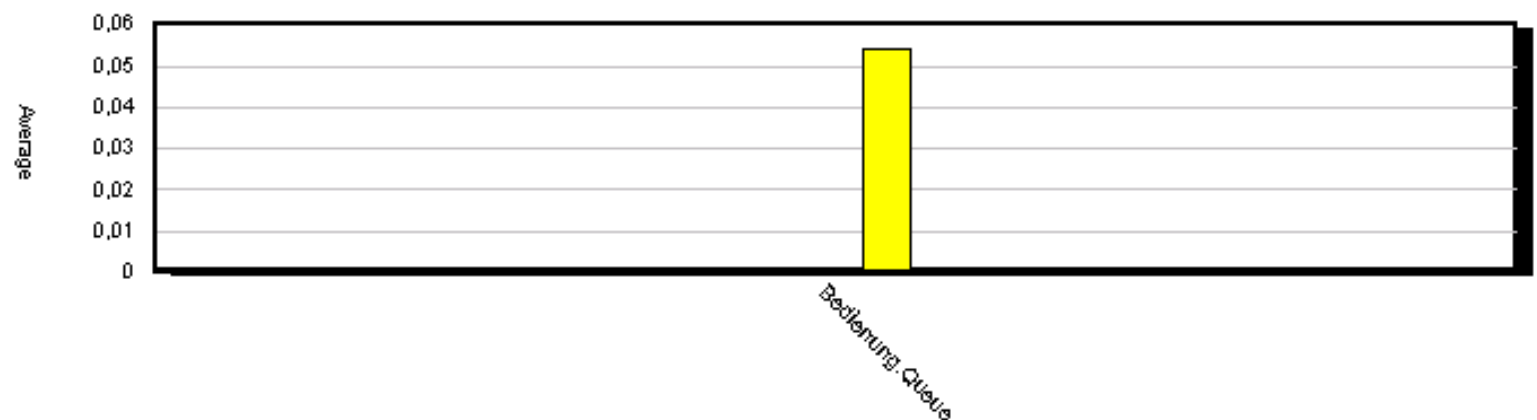
	Average	HalfWidth	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Bedienung.Queue	0.02629859	0	0.01778652	0.04508268	0.00	0.8520



Other

Number Waiting

	Average	HalfWidth	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Bedienung.Queue	0.05420458	0	0.03668619	0.0925	0.00	5.0000



Weitere Module

- Im Folgenden werde weitere Module für den Aufbau eines Simulationsmodells aufgelistet.
- Nähere Information zu den einzelnen Komponenten erhält man im Menü *Help->Arena Help Topics->Using the Basic Process Panel->Flowchart Modules*.
- *Decide Module*: Repräsentiert einen Entscheidungsprozess in einem Modell (z.B. gehen Kunden mit der Wahrscheinlichkeit (WK) p zum Schalter a und mit WK $1-p$ zum Schalter b).
- *Batch/Separate Module*: Ströme von *Entities* können zusammengefaßt werden (Batch) und später wieder getrennt werden (Separate).

Weitere Module

- *Assign Module*: Weist Variablen Werte zu (z.B. könnte man die Variable *Timestamp* einführen, die an einer Station des Modells den Zeitpunkt speichert, an dem ein Objekt diese passierte).
- *Record Module*: Sammelt während der Simulation Zwischenergebnisse.

Ein fortgeschrittenes Projekt

- Problemstellung:

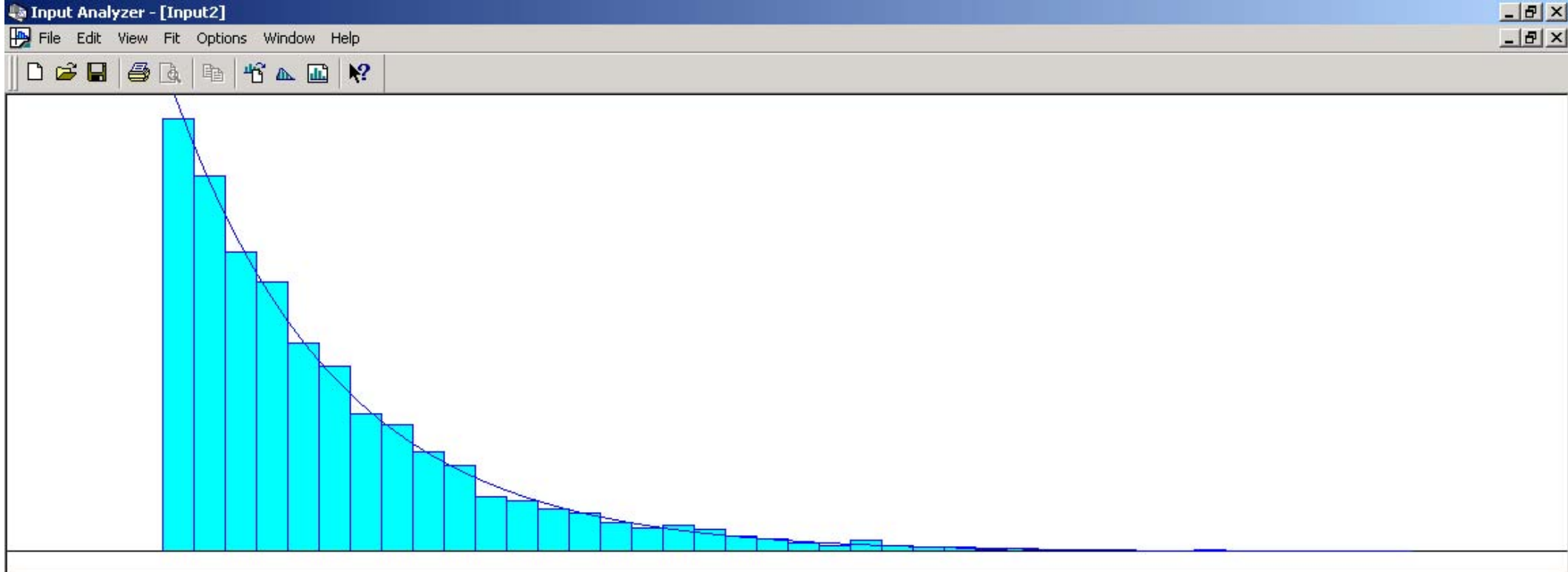
Kunden betreten eine Videothek und entscheiden, welches Video sie ausleihen wollen. Danach kaufen sie eventuell noch etwas Popcorn und bezahlen.

- Ankunftsprozess:

- Wir wissen nicht, in welchen Zeitabständen Kunden ankommen, haben aber die Meßdaten einer Woche dieser Zeitabstände.
- Der *Inputanalyzer* (Menü *Tools*) sucht uns eine Verteilung, die diese Zwischenankunftszeiten gut modelliert.
- Wir starten im *Inputanalyzer* mit *New* eine neue Analyse und wählen unter *File->Data File->Use Existing* die Datei mit unseren Meßdaten aus.

Ein fortgeschrittenes Projekt

- Die Meßdaten können z.B. in einer Textdatei (durch ein Leerzeichen getrennt) stehen.
- Nach dem Einlesen erscheinen die Daten als Balkendiagramm im Hauptfenster.
- Mit dem Befehl *Fit->Fit All* paßt Arena den Daten eine Verteilung an.
- Im unteren Fenster wird die am besten passende Verteilung aufgeführt (hier eine Exponentialverteilung mit Erwartungswert 1.98).
- Man kann sich im Menü Fit auch auf eine spezielle Verteilung festlegen.



Distribution Summary

Distribution: Exponential
Expression: $\text{EXP0}(1.98)$
Square Error: 0.000159

Chi Square Test

Number of intervals	= 28
Degrees of freedom	= 26
Test Statistic	= 26.2
Corresponding p-value	= 0.458

Kolmogorov-Smirnov Test

Test Statistic	= 0.00945
Corresponding p-value	> 0.15

Data Summary

Number of Data Points	= 5000
Min Data Value	= $5.24\text{e-}005$
Max Data Value	= 14.7
Sample Mean	= 1.98

For Help, press F1

Ein fortgeschrittenes Projekt

- Die Verteilung sollte „wahre Verteilung“ möglichst gut modellieren, daher müssen die statistischen Tests (z.B. Chi Square Test) möglichst kleine Test-Statistiken haben.
- Im Hauptfenster wird die Funktion der angepaßten Verteilung zusätzlich zu den Meßdaten eingezeichnet. Daran läßt sich grob abschätzen, ob die Verteilung die Daten gut modelliert.
- In unserem Beispiel wissen wir nun, daß wir im *Create Module* unter *Type Random(Expo)* und unter *Value 1.98* eingeben müssen.

Ein fortgeschrittenes Projekt

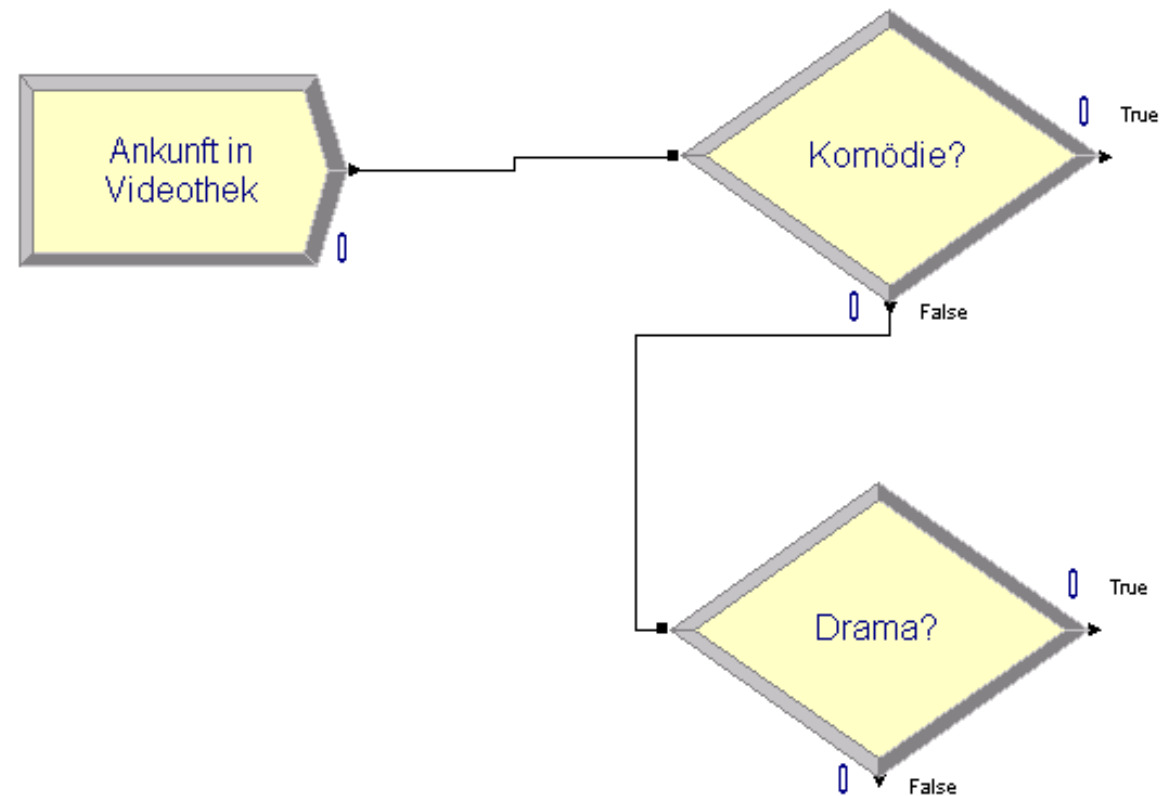
- Damit haben wir das passende *Create Module* für unser Modell.
- Als nächstes muß der Kunde auswählen, ob er eine Komödie, ein Drama oder einen Actionfilm sehen will. Das modellieren wir mit zwei *Decide Modules*.
- Die WK, daß der Kunde sich für eine Komödie entscheidet sei 50% und für ein Drama oder ein Actionfilm jeweils 25%.
- Wir beginnen mit dem ersten *Decide Module*, bei dem sich für oder gegen eine Komödie mit einer WK von jeweils 50 % (*Type* ist *2-Way-Chance* und *Percent True* ist 50%). Es wird mit dem *Create Module* verbunden.

Ein fortgeschrittenes Projekt

- Ein zweites *Decide Module*, bei dem sich für oder gegen ein Drama entschieden wird, erhält ebenfalls die Werte *Type: 2-Way-Chance* und *Percent True: 50%*.

!Achtung! Es handelt sich hier um bedingte WK, denn wenn man sich schon gegen die Komödie entschieden hat, bleibt eine 50%ige WK für jede der zwei verbleibenden Möglichkeiten, da diese ja gleich wahrscheinlich sind.

- Nun muß mit dem *Connect*-Button das untere Ende des ersten *Decide Module* (das *False*-Ende) mit dem Eingang (linkes Ende) des zweiten *Decide Module* verbunden werden.

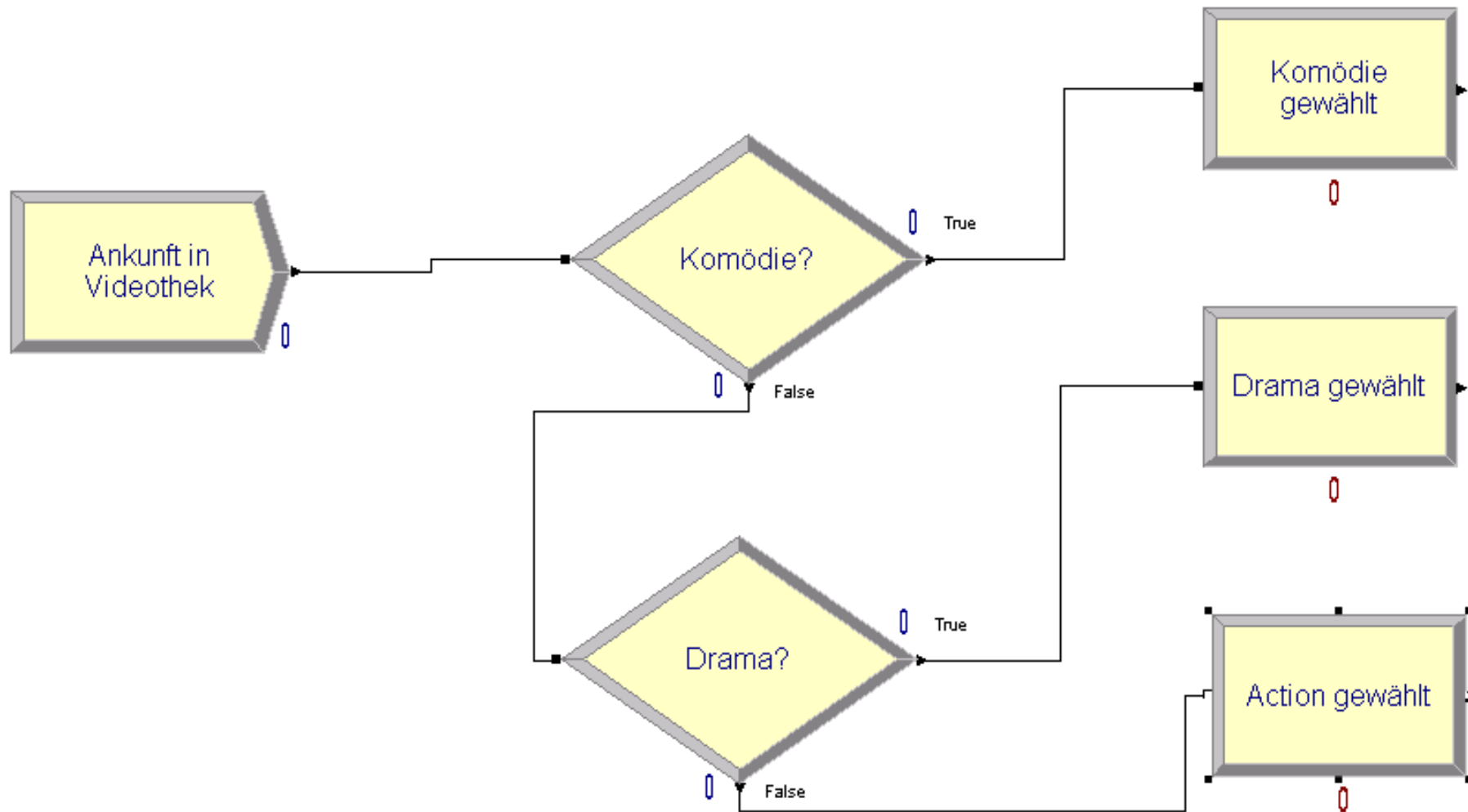


Decide - Basic Process

	Name	Type	Percent True
1	Komödie?	2-way by Chance	50
2	Drama?	2-way by Chance	50

Ein fortgeschrittenes Projekt

- Jetzt brauchen wir drei *Process Modules*, um die Dauer für das Aussuchen des Filmes zu simulieren. Diese Dauer ist für jedes Genre verschieden.
- Mit dem *Connect*-Button werden wieder die entsprechenden Endpunkte der Module verbunden.
- Wir wählen für jedes der drei *Process Modules* unter *Action* den Wert *Delay*, da wir hier keine Warteschlange haben.
- Weiterhin sei die Zeit zum Aussuchen des Filmes gleichverteilt (*uniform*) auf dem Intervall $[1,2]$ (in Minuten) für die Komödien, gleichverteilt auf dem Intervall $[2,3]$ für die Dramen und gleichverteilt auf dem Intervall $[3,4]$ für die Actionfilme.

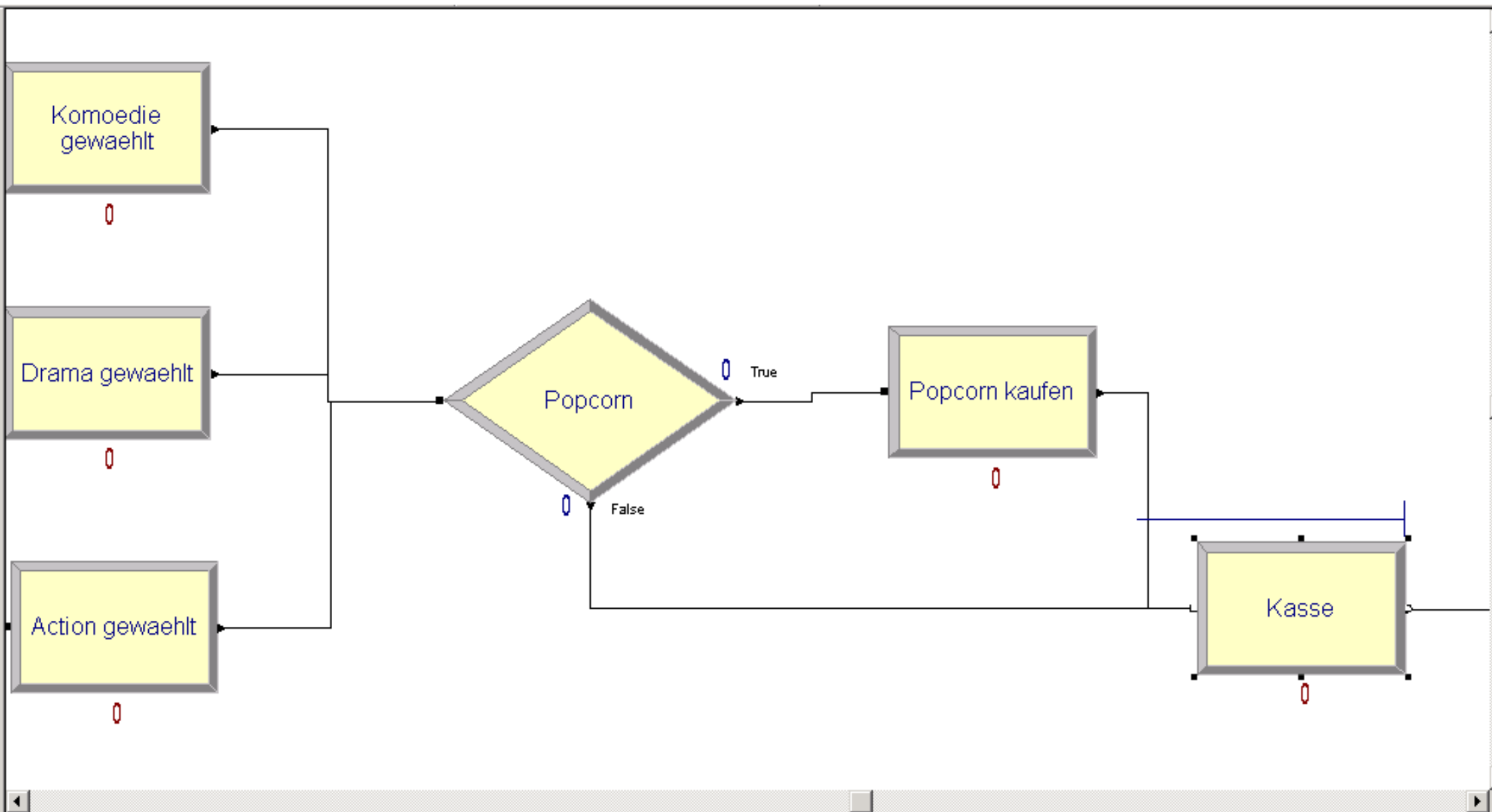


Process - Basic Process

	Name	Type	Action	Delay Type	Units	Allocation	Minimum	Maximum	Report Statistics
1	Komödie g	Standard	Delay	Uniform	Minutes	Value Added	1	2	<input checked="" type="checkbox"/>
2	Drama ge	Standard	Delay	Uniform	Minutes	Value Added	2	3	<input checked="" type="checkbox"/>
3	Action ge	Standard	Delay	Uniform	Minutes	Value Added	3	4	<input checked="" type="checkbox"/>

Ein fortgeschrittenes Projekt

- Als nächstes haben die Kunden noch die Möglichkeit Popcorn zu kaufen. Ein Decide Module modelliert diese Möglichkeit (25% der Kunden kaufen Popcorn). Dessen linkes Ende wird mit jedem der drei Process Modules verbunden.
- Die Kunden, die sich für Popcorn entschieden haben, werden noch für eine Zeitdauer aufgehalten, die dreiecksverteilt (*Triangular*) ist auf $[0.5, 2]$ mit dem Erwartungswert 1.
- Nun müssen die Kunden an der Kasse ihren Film und eventuell das Popcorn bezahlen. Wir brauchen also ein *Process Module* mit *Action: Seize Delay Release*. Als Wartezeit wählen wir eine Gleichverteilung auf $[2, 5]$. Das Module hat eine *Resource*.



Process - Basic Process

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics
1	Komoedie gewae	Standard	Delay	Medium(2)	0 rows	Uniform	Minutes	Value Added	1	1	2	<input checked="" type="checkbox"/>
2	Drama gewaehlt	Standard	Delay	Medium(2)	0 rows	Uniform	Minutes	Value Added	2	1	3	<input checked="" type="checkbox"/>
3	Action gewaehlt	Standard	Delay	Medium(2)	0 rows	Uniform	Minutes	Value Added	3	1	4	<input checked="" type="checkbox"/>
4	Popcorn kaufen	Standard	Delay	Medium(2)	0 rows	Triangular	Minutes	Value Added	.5	1	2	<input checked="" type="checkbox"/>
5	Kasse	Standard	Seize Delay Release	Medium(2)	1 rows	Uniform	Minutes	Value Added	2	1	5	<input checked="" type="checkbox"/>

Ein fortgeschrittenes Projekt

- Als letztes fügen wir noch ein *Dispose Module* ein, das das Verlassen der Videothek modelliert.
- Im *Run Setup* wählen wir 5 Replikationen mit jeweils 800 Minuten.
- Startet man nun die Simulation, so erscheint während der Simulation eine Fehlermeldung:

ERROR:

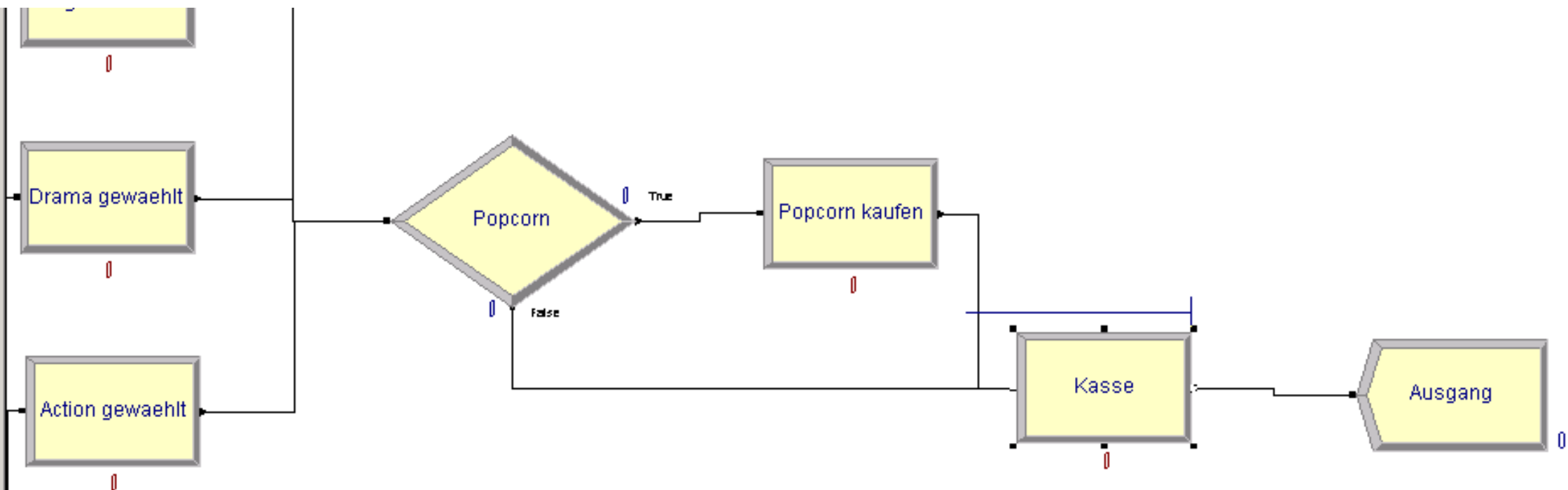
Entity: 100

A runtime error was detected...

- Der Grund ist die Warteschlange für der Kasse. Die Kapazität ist auf 100 Kunden begrenzt. Wir müssen unseren Kassierer also schneller arbeiten lassen.

Ein fortgeschrittenes Projekt

- Wir haben nun zwei Möglichkeiten, den „Fehler“ zu beheben (Sollte unsere Videothek wirklich mit diesen Bedien- und Ankunftszeiten arbeiten, wäre sie schnell völlig überfüllt):
 1. Die Zwischenankunftszeit im *Create Module* vergrößern, so daß die Kunden in größeren Zeitabständen ankommen.
 2. Den Kassierer „schneller“ machen, bzw. die Bedienzeit im *Process Module* verkleinern.
- Wir wählen hier die zweite Variante und lassen wählen für die Bedienzeit an der Kasse die Gleichverteilung aus $[0.5, 3]$.



Process - Basic Process

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report
1	Komoedie gewaehlt	Standard	Delay	Medium(2)	0 rows	Uniform	Minutes	Value Added	1	1	2	<input checked="" type="checkbox"/>
2	Drama gewaehlt	Standard	Delay	Medium(2)	0 rows	Uniform	Minutes	Value Added	2	1	3	<input checked="" type="checkbox"/>
3	Action gewaehlt	Standard	Delay	Medium(2)	0 rows	Uniform	Minutes	Value Added	3	1	4	<input checked="" type="checkbox"/>
4	Popcorn kaufen	Standard	Delay	Medium(2)	0 rows	Triangular	Minutes	Value Added	.5	1	2	<input checked="" type="checkbox"/>
5	Kasse	Standard	Seize Delay Release	Medium(2)	1 rows	Uniform	Minutes	Value Added	0.5	1	3	<input checked="" type="checkbox"/>

Ein fortgeschrittenes Projekt

- Simulieren wir mit diesen Einstellungen, erhalten wir z.B. für die durchschnittliche Wartezeit an der Kasse 6 Minuten und 11 Sekunden. Es warten im Mittel 3,14 Kunden in der Schlange (Verzeichnis *Queue* des Reports).
- Der Angestellte ist zu ca. 88 % ausgelastet (Verzeichnis *Resource*).
- Ein Kunde verbringt im Mittel 10 Minuten und 37 Sekunden in der Videothek.
- Diese Ergebnisse beziehen sich nur auf **diese** Simulation. Da wir aber immerhin 5 lange Simulationsläufe gemacht haben, können wir davon ausgehen, daß sie nur minimal von den „wahren“ Mittelwerten abweichen.

Hilfe in Arena

- Pdf-Dokument im Startmenü unter *Arena-> Online Books -> Arena Standard Edition User's Guide* (mit einer guten Einführung).
- Hilfemenü im Arenaprogramm -> *Arena Help Topics* (ähnlich zum Pdf, aber ausführlicher und mit *Suche/Index*).
- Arena-Examples im Installationsordner -> *Examples* (größtenteils sehr fortgeschrittene Projekte)
- *Smart File Library* (im Ordner *Smarts* oder über *Arena Help Topics -> Exploring Arena -> Learning Arena with Smart Files* (sehr empfehlenswert, in einfachen Modellen werden alle Konzepte erklärt).