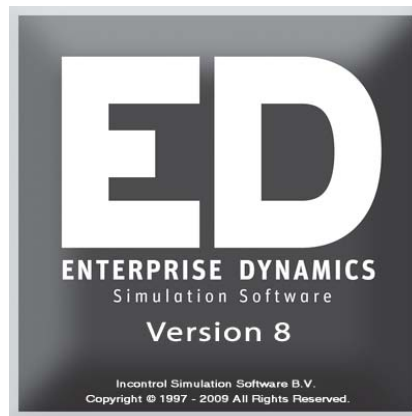


TUTORIAL ED 8



Copyright © 2009, Incontrol Simulation Solutions
Papendorpseweg 77
3528 BJ Utrecht
The Netherlands
www.IncontrolSim.com



VORWORT

Um ein Auto fahren zu dürfen, ist der Besitz einer Fahrerlaubnis gesetzlich vorgeschrieben. Sie ist der Nachweis für eine erfolgreich absolvierte Ausbildung, wobei neben der Beherrschung des Fahrzeugs auch die Regeln für die sichere Nutzung im Umgang mit anderen Verkehrsteilnehmern geprüft wurden. Wenige werden die Illusion haben, dass das Lesen des mitgelieferten Hefts im Handschuhfach für die Nutzung eines Autos ausreichend ist.

Für Simulationssoftware mit ihren umfassenden Möglichkeiten gibt es keine Fahrerlaubnis, aber auch hierfür ist Training, sowohl für die Beherrschung der Software als auch für die vertretbare Anwendung bei Simulationsproblemen, erforderlich. Und in Analogie zum mitgelieferten Heft bei einem neuen Auto: lediglich die Anleitung durchzuackern ist selten ausreichend.

Enterprise Dynamics, (ED) von Incontrol Simulation Solutions ist eine Software für diskrete Simulation. Sie wird mit einer englischsprachigen Anleitung in mehreren Teilen geliefert. Von Anwendern auf Anfängerniveau – und insbesondere aus dem Bildungssektor – kam die Nachfrage nach einer didaktischer gestalteten Anleitung, mit der diese Anwendergruppe zügig und selbstständig die ersten Prinzipien von ED erlernen könnten.

Dies führte zu diesem Tutorial und war ein erster Anlauf, diesem Bedarf zu entsprechen. Das Tutorial stellt einen wesentlichen Bestandteil der Educational Suite dar: die Gesamtheit von Software und modernem Ausbildungsmaterial mit dem Ziel, Studenten herauszufordern und Dozenten ein flexibles und vielseitiges Ausbildungsprogramm anzubieten.

Andere Komponenten dieser Educational Suite sind unter anderem eine Sammlung von Fallstudien (Problem, Ausarbeitung gemäß einer Simulationsmethodik, dazugehörige ED-Modelle) und eine ED Studentenversion.

In Zukunft wird dieses Tutorial wiederholt Anpassungen und Ergänzungen erfahren, was allein schon für neue ED-Versionen erforderlich ist.

Alle Beurteilungen bezüglich dieses Tutorials oder der Software, Verbesserungsvorschläge etc. sind willkommen unter education@IncontrolSim.com. Diese eMail-Adresse gilt nicht für inhaltliche Fragen zu ED: hierfür ist in erster Linie der Dozent der Ansprechpartner! Der Dozent kann sich anschließend (sofern erforderlich) an das Helpdesk von Enterprise Dynamics wenden.

Nienke Valkhoff, Manager Training & Education

Utrecht, Dezember 2009

INHALTSVERZEICHNIS

1	Die gestaltung des Tutorials	4
1.1	Der didaktische Ansatz.....	4
1.2	Anmerkung	4
1.3	Der Aufbau des Tutorials	5
1.4	Simulation lernen	5
2	Die Hintergründe von ED	7
3	Die erste Begegnung mit ED.....	8
3.1	Enterprise Dynamics starten	8
3.2	Erläuterung der Fensterkomponenten	9
3.3	Die Menüstruktur	11
3.4	Die Struktur der Bibliothek und des Modells	12
4	Die Grundlagen der Modelgenerierung	13
4.1	Atome in das Modell ziehen	13
4.2	Die Kanäle.....	18
5	Ergebnisse anschauen	24
5.1	Methoden zur Messung der Ergebnisse	26
5.2	Das Messen der Ergebnisse	27
6	Spielen mit Strategien	41
6.1	Die Anpassung der Inputstrategie.....	42
6.2	Das Anpassen der Queue Discipline	43
6.3	Das Anpassen des Send to Statements	44
7	Weitere Atome: von Assembler bis Unpack	46
8	Enterprise Dynamics und Excel	51
8.1	Die Bank.....	51
8.2	Die Anbindung an Excel	53
8.3	Abspeichern von Daten in Excel.....	55
8.4	Import von Daten aus Excel.....	56
8.5	Troubleshooting	57
8.6	To pool or not to pool?.....	58
	Anlage 1 Die Menüstruktur	
	Anlage 2 Eine Beschreibung der wichtigsten Atome	
	Anlage 3 A first guide to 4DScript	

1 DIE GESTALTUNG DES TUTORIALS

1.1 Der didaktische Ansatz

Dieses Tutorial vermittelt einem Anwender auf Anfängerniveau die Grundprinzipien des Simulationspakets Enterprise Dynamics, nachfolgend „ED“ abgekürzt.

Den Anfang in Abschnitt 3 bildet ganz einfach das Startfenster von ED. Von dort aus lernt man ED über ein schrittweises Vorgehen zunehmend kennen. Dies geschieht zumeist anhand (kurzer) Fallbeispiele: kleine Probleme in einem bestimmten Kontext. Durch den Schritt für Schritt erfolgenden Aufbau eines dazugehörigen Modells oder gerade das Experimentieren damit werden die Funktionalitäten von ED vor dem Hintergrund eines Anwendungsbeispiels deutlich.

Dieses Vorgehen bietet eine Reihe von Vorteilen:

1. der Anwender lernt ED über das Modellieren eines Simulationsproblems kennen, was der Anwendung des Pakets in der Praxis entspricht
2. Funktionalitäten eines Programms kann man sich besser merken, wenn sie an Anwendungsbeispiele gekoppelt werden
3. der Anwender kann auf mehrere Lösungen in Beispielen zurückfallen, wenn er mit einem solchen Problem konfrontiert wird.

Kurzum, dieses Tutorial versucht nicht nur zu vermitteln, *was* eine Komponente der Software macht, sondern auch, *wie* und *wann* sie genutzt wird. Einige Komponenten, insbesondere die Übersichten in den Anlagen, fungieren daneben als Nachschlagewerk.

Um Missverständnisse zu vermeiden: es ist keinesfalls so, dass diese Anleitung das (englischsprachige) Manual – das unter der Hilfefunktion zu finden ist – überflüssig macht. Anwender, die beim Generieren von Simulationsmodellen weiterkommen wollen, werden gerade häufig das Manual zu Rate ziehen, da dieses eine vollständige Darstellung aller Funktionalitäten der Software beinhaltet.

1.2 Anmerkung

Wegen der niederländischsprachigen Gestaltung wurden die englischen Begriffe mitunter übersetzt, häufig entschied man sich aber dafür, Begriffe nicht zu übersetzen, einerseits aufgrund des Fehlens eines niederländischen Äquivalents und andererseits, weil dies das Erlernen von ED erschweren würde (vor allem in Fällen englischer Menüoptionen).

Wenn auf ein Untermenü verwiesen wird, so geschieht dies durch das Symbol |. So verweist File | Preferences beispielsweise auf das Untermenü Preferences (hierunter fallen einige Standardeinstellungen), das unter dem File-Menü zu finden ist. Das Betonen wesentlicher Aspekte erfolgt durch **Fett**- oder *Kursiv*-Schreibung. Die Fallbeispiele und die dazugehörigen Fragen werden in diesem Schrifttyp dargestellt. Wenn in einem Text 4DScript Code enthalten ist, wird dieser Schrifttyp verwendet.

Ein Warnhinweis beginnt mit *Achtung!* Ein wichtiger Hinweis, der auch in vielen anderen Fällen Anwendung finden kann, mit *Tipp*.

1.3 Der Aufbau des Tutorials

Nach einer kurzen allgemeinen Einführung zum Programm ED in Abschnitt 2 beginnt in Abschnitt 3 die Vorstellung des Pakets selbst in Form eines Überblicks über die Menüstruktur und die Komponenten des Startfensters von ED.

Die Grundprinzipien des Pakets werden in Abschnitt 4 anhand eines einfachen Warteschlangenproblems mit einem Schalter demonstriert. Dies wird später um zwei weitere Schalter ergänzt, um das Konzept der so genannten Kanäle zu zeigen. Die genutzten Atome sind Source, Queue, Server und Sink.

In Abschnitt 5 stehen die verschiedenen Möglichkeiten zum Betrachten der Ergebnisse einer Studie im Mittelpunkt. Dies erfolgt anhand eines Fallbeispiels in einer Zimmerei, wo unter anderem die Arbeit mit Batches verdeutlicht wird. Der Anwender lernt dabei die neuen Komponenten Monitor, Summary Report, verschiedene Grafiken und den Experiment Wizard kennen. Daneben üben wir auf Elementarebene die Schritte in einer Simulationsstudie von Modellgenerierung, Validierung, Experimentgestaltung und Analyse der Ergebnisse.

In Abschnitt 6 liegt der Schwerpunkt auf der Betrachtung der vorprogrammierten Möglichkeiten, mit denen Produkte Zugang zu einem folgenden Atom erhalten (Input-Strategie), in einer Warteschlange platziert werden (Wartedisziplin) oder nach Bearbeitung weitergeschickt werden (send to).

Abschnitt 7 führt acht neue Atome ein, darunter den Assembler und den Conveyor, während Abschnitt 8 der Anbindung von ED an Excel gewidmet ist. Dieser Abschnitt ist zugleich ein Start in 4DScript, u.a. über die wesentlichen Label.

Die Anlagen haben eine wichtige Funktion als Nachschlagewerk. In Anlage 1 steht eine kurze Erläuterung zur Menüstruktur, während in Anlage 2 die wichtigsten Atome vollständig beschrieben werden. Anlage 3, A first guide in 4DScript, ist englischsprachiger erster Schritt in die ED zugrunde liegende Programmiersprache.

1.4 Simulation lernen

Simulation ist keine einfache Technik: ein sinnvoller Einsatz von Simulationsprogrammen erfordert Kenntnisse bezüglich der theoretischen Hintergründe der (diskreten) Simulation wie zum Beispiel Kenntnisse im Hinblick auf Wahrscheinlichkeitsverteilungen und den Prozess der Modellierung wie z.B. Validierungstechniken oder die Gestaltung von Experimenten. Obwohl das Tutorial auch auf diesem Gebiet Wissen vermittelt, ist es seiner Art nach *kein* Simulationslehrbuch. Hierfür wird auf bestehende Bücher oder Diktate zum Thema diskrete Simulation verwiesen.

Die Entwicklung eines Gefühls für Simulation und die Generierung von (Simulations-) Modellen im Besonderen ist vor allem eine Frage intensiven Übens. Eine Studentenversion 6.0 – geeignet für die Speicherung von Modellen bis zu 30 Atomen – steht unter der Internetadresse www.enterprisedynamics.com zum Download bereit. Es ist nicht möglich, größere Modelle zu generieren oder zu importieren.

Jedem, der im Besitz eines Computers mit den dort aufgeführten Spezifikationen ist, wird empfohlen, diese Version auf dem Heimrechner zu installieren. Achten Sie bitte darauf, dass die

kommerzielle Nutzung oder die Verwendung im Ausbildungsbereich ohne den Besitz einer Klassenraumlizenz nicht gestattet ist. Schauen Sie auf der Website regelmäßig nach neuen Entwicklungen. Hier wird in Kürze unter anderem der erste Teil dieses Tutorials in verschiedenen Sprachen zum Download bereitstehen.

Die Beispielmodelle in diesem Tutorial sind im Lieferumfang der Studentenversion enthalten. Verwenden Sie bei Zweifeln bezüglich der Korrektheit die eigenen Modelle.

2 DIE HINTERGRÜNDE VON ED

Enterprise Dynamics ist eine objektorientierte Software zum Modellieren, Simulieren und Visualisieren dynamischer Prozesse. Aus Standardbibliotheken können Anwender Komponenten – Atome genannt – zur Generierung Ihres Modells laden. ED basiert auf der Vorstellung von Atomen als Bausteine jedes Modells.

Ein Atom kann eine Maschine, einen Schalter oder ein Produkt darstellen, aber auch einen nicht-physischen Charakter tragen, wie zum Beispiel eine Grafik. Im Interesse der Unterscheidung bei der Art von Atomen werden wir von *Basisatomen* (fünf häufig genutzte Atome: Product, Source, Sink, Server und Queue), *Transportatomen* (transportbezogen), *Experimentatomen* etc. sprechen.

Aufgrund der offenen Struktur von ED kann der (fortgeschrittene) Anwender mit dem Atom Editor seine eigenen Atome generieren und verwenden. Denken Sie zum Beispiel an eine Maschine mit sehr spezifischen Eigenschaften. Momentan enthält die ED-Standardbibliothek etwa 100 Atome, deren Zahl jedoch stetig steigt. Der Anwender auf Anfängerniveau wird für seine Anwendungsbeispiele mit etwa 30 häufig genutzten Atomen auskommen.

Die Atome sind folglich vordefinierte Bausteine zur raschen Erstellung von Modellen und Ausführung einer Studie. Daneben ist in ED eine eigene Programmiersprache integriert, 4DScript genannt, mit der spezifische Aspekte aus der Wirklichkeit im Modell verarbeitet werden können. Diese Sprache umfasst momentan etwa 1100 Begriffe. In diesem Tutorial wird hierauf nur kurz eingegangen.

Aufgrund des offenen Charakters von ED kann der Anwender das Paket selbst erweitern oder ein eigenes Layout wählen. Incontrol liefert selbst ein paar dieser so genannten Suites:

- die Logistics Suite für Produktion, Material-Handling und Distribution
- die Airport Suite für Flughäfen
- Die Educational Suite für den Bildungssektor

Dieses Tutorial kann für die Logistics Suite genutzt werden und ist als Produkt Bestandteil der Educational Suite.



Incontrol Simulation Solutions als Unternehmen ist ein Simulation Solution Provider: neben der Generierung, dem Bau und dem Verkauf der Enterprise Dynamics Software erbringt das Unternehmen auch Dienstleistungen im Bereich der Simulation, wie zum Beispiel Trainings und Consultancy. Die Consultancy-Tätigkeiten beziehen sich sowohl auf die Ausführung von Simulationsstudien wie auch auf die Anwendungsentwicklung.

3 DIE ERSTE BEGEGNUNG MIT ED

3.1 Enterprise Dynamics starten

Enterprise Dynamics kann über das Startmenü gestartet werden. Beim Start erscheint zunächst ein Splash-Screen (Abbildung 3-1), wonach mitunter die Frage folgt, welche Anwendung gestartet werden soll (Abbildung 3-2). Hier ist eine .app Datei auszuwählen.

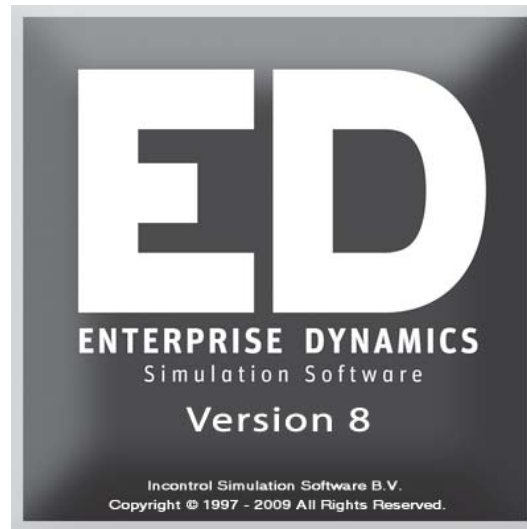


Abbildung 3-1: Splash-Screen

Häufig wird Enterprise Dynamics diese Auswahl der Anwendung selbst vornehmen, in dem Fall bekommt der Anwender kein Eingabefenster zu sehen.

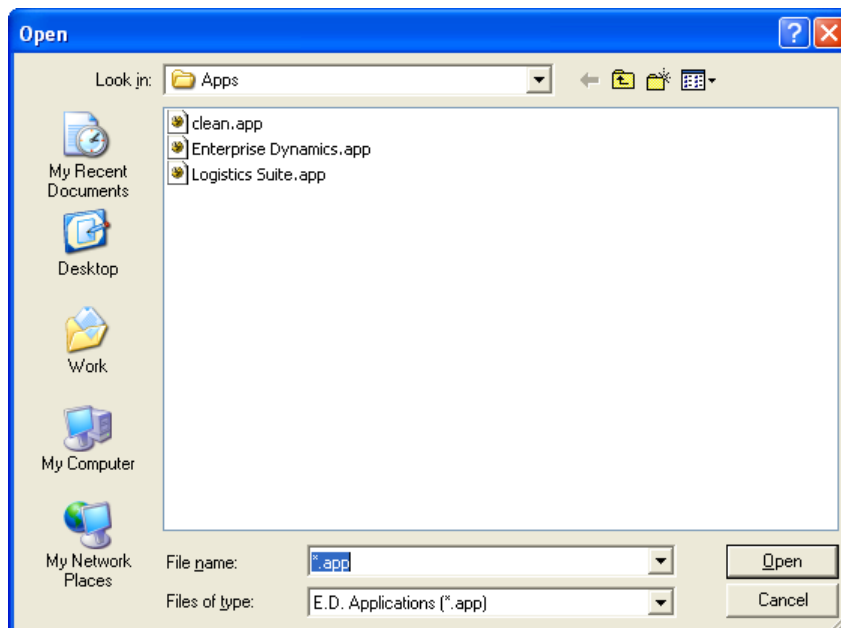


Abbildung 3-2: Auswahl der Anwendung

In den Anwendungsdateien sind die Einstellungen für das Programm Enterprise Dynamics selbst enthalten. Unter anderem ist definiert, welche Menüs der Anwender zu sehen bekommt und welche Atome sofort verfügbar sind. Diese Anwendungsdateien können vom Anwender angepasst werden.

3.2 Erläuterung der Fensterkomponenten

Sobald Enterprise Dynamics vollständig gestartet wurde, müsste das Startfenster (mit Ausnahme der Ovale) ungefähr wie in Abbildung 3-3 aussehen.

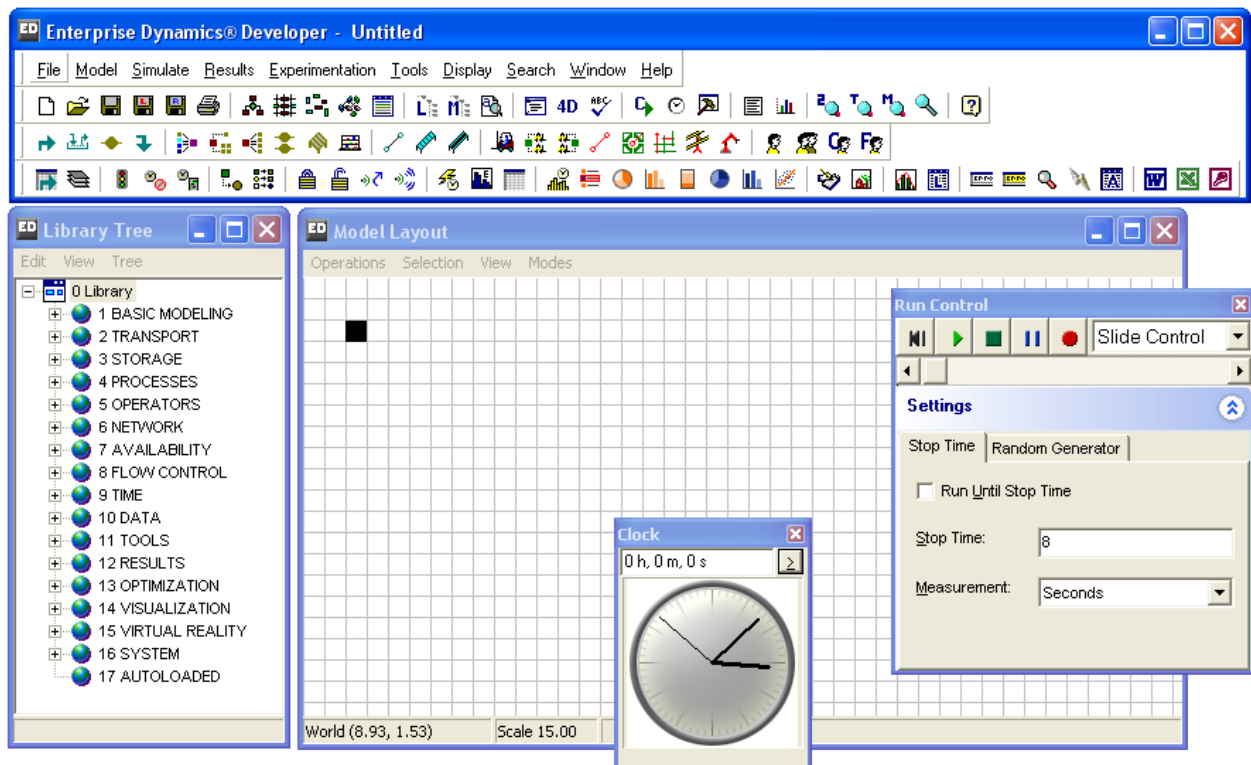


Abbildung 3-3: Aufbau Anfangsfenster Enterprise Dynamics

Dieses Fenster ist aus folgenden Komponenten aufgebaut:

- Eine *Menüleiste*: hiermit können unter anderem Dateien geöffnet und gespeichert werden. In Abschnitt 3.3 wird weiter auf die Menüstruktur eingegangen.
- Die *Speed Buttons*: mit Hilfe dieser Buttons können bestimmte Atome schnell im Modell platziert und Befehle an Enterprise Dynamics gegeben werden (zum Beispiel Speichern eines Modells). Wenn ein Speed Button angeklickt wird, erscheint das Atom automatisch im Modell oder es wird eine Aktion gestartet.

- Die *Bibliothek*: hier sind alle Atome enthalten, die ein Anwender im Modell platzieren kann. Jedes Atom hat eine bestimmte Funktion, und durch die Kombination der richtigen Atome ist es möglich, einen Betriebsprozess in Enterprise Dynamics nachzubauen (zu „modellieren“).
In Abschnitt 4 wird die Generierung von Modellen behandelt.
- Das *Modellfenster*: in diesem Teil des Fensters werden die Modelle generiert.
- Die *Run Control*: hiermit wird die Simulation des entworfenen Modells gestartet und kann die Geschwindigkeit der Simulation eingestellt werden.

3.3 Die Menüstruktur

Die Funktion und das Aussehen der Menüs entspricht den Menüs in anderen Windows Anwendungen, wie zum Beispiel Word und Excel. In der nachfolgenden Tabelle wird zu den am häufigsten verwendeten Menüoptionen eine kurze Erläuterung gegeben.

Die Menüleiste enthält das Hauptmenü. Dessen Komponenten sind:

File	Für das Anlegen, Öffnen oder Speichern von Dateien oder die Einstellung von Standardfunktionalitäten wie Drucken oder Dateiverzeichnis
Model	Für die Generierung und Darstellung von Modellen
Simulate	Für die tatsächliche Ausführung einer einzelnen Simulation
Results	Für Reports und Grafiken bezüglich eines einzelnen Simulationsdurchlaufs
Experimentation	Für den Entwurf, die Ausführung und die Evaluierung eines Experiments mit <i>mehreren</i> Simulationsdurchläufen
Tools	Umfasst insbesondere Hilfsmittel für den Bau von Atomen, z.B. diverse Editoren
Display	Für die Wahl der Modelldarstellung in 2D oder 3D
Window	Umfasst diverse Fenster, wie z.B. eine Übersicht von 4D-Script oder Icons
Help	Umfasst das vollständige Manual sowie Firmen- und Versionsinformationen

Achten Sie bitte darauf, dass die Optionen Model, Simulate und Results den nacheinander folgenden Schritten einer Simulationsstudie folgen!

Hinter jedem Hauptmenüpunkt verbergen sich Untermenüs. In Anlage 1 finden Sie eine kurze Erläuterung dieser Struktur mit einer Beschreibung aller Untermenüs. Die *kursiv* dargestellten Menüoptionen sind lediglich für den fortgeschrittenen Anwender von Bedeutung und können von Anfängern übersprungen werden. In der Anlage entschied man sich für eine vollständige Übersicht, da dieser Bestandteil des Tutorials auch eine Funktion als Nachschlagewerk erfüllt. Lesen Sie Anlage 1 bitte kurz durch, um einen ersten Eindruck der Programmstruktur zu erhalten. Ein fundiertes Wissen bezüglich dieses Aufbaus ist insbesondere beim späteren selbständigen Generieren von Modellen hilfreich.

3.4 Die Struktur der Bibliothek und des Modells

In ED nutzen wir eine Baumstruktur für die Visualisierung der Ordnung von Atomen. Hiermit zeigen wir auf, welche Atome welche anderen Atome enthalten. So gibt der „main tree“ einen vollständigen Überblick über die Anwendung, die Bibliothek und das geöffnete Modell:

Zwei andere wichtige Baumstrukturen sind:

- Der Library Tree (siehe Abbildung 3-4), hierin sind alle Atome enthalten, die der Anwender im Modell platzieren kann. Die Atome sind in Gruppen unterteilt. So gibt es unter anderem eine Gruppe mit Conveyors (Förderern) und eine Gruppe mit Operators(Werkern). Wenn Sie ein Atom auswählen und zum Modell-Fenster ziehen ("Model Layout"), wird das Atom im Modell platziert.
- Der Model Tree, hierin sind alle Atome enthalten, die im Modell platziert wurden.

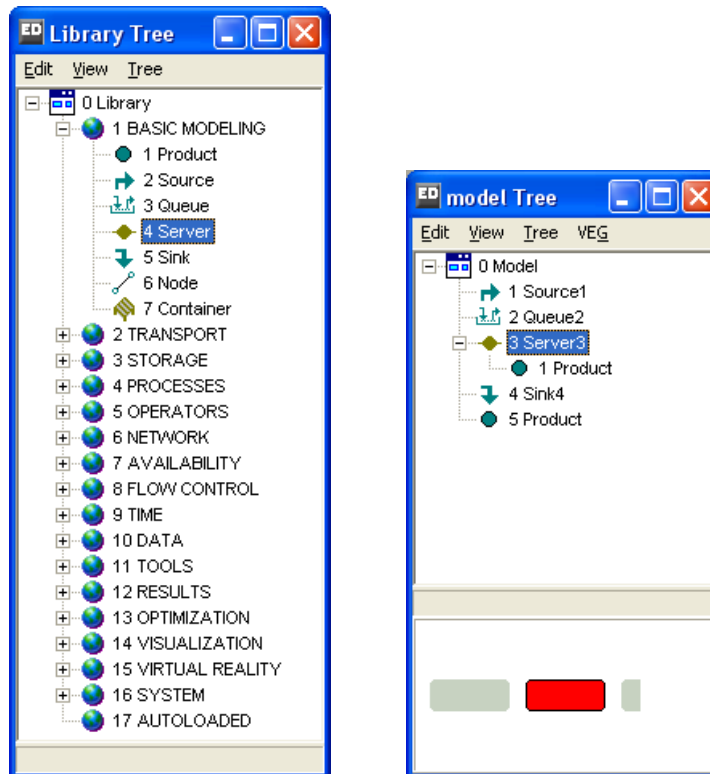


Abbildung 3-4: der allgemeine Library Tree und der Model Tree von Casus 1

Zwischen den beiden Teilen der Bibliothek kann gewechselt werden, indem Sie im Menü Model die Menüoption Library Tree oder Model Tree wählen. Daneben können die Speed Buttons genutzt werden. Abbildung 3-5 zeigt, welche Buttons dann verwendet werden müssen.



Abbildung 3-5: Speed Buttons für den Library Tree und den Model Tree

4 DIE GRUNDLAGEN DER MODELGENERIERUNG

Nachdem im vorigen Abschnitt lediglich die Theorie im Hinblick auf Enterprise Dynamics erörtert wurde, wird in diesem Abschnitt mit der Generierung eines einfachen Modells in Enterprise Dynamics begonnen. Übrigens liegt der Schwerpunkt hierbei auf dem Erlernen der Arbeit mit ED und nicht auf der vollständig vertretbaren Ausführung einer Simulationsstudie. Zum Ende dieses Abschnitts sollte der Anwender in der Lage sein, ein Modell zu entwickeln, in dem mehrere Maschinen parallel genutzt werden.

4.1 Atome in das Modell ziehen

Der erste Schritt beim Generieren eines Modells ist die Platzierung der richtigen Atome im Modell. In diesem Abschnitt beginnen wir mit dem Bau eines einfachen Modells, das folgende vier Komponenten umfasst (siehe Abbildung 4-1):

- Source, dieses Atom sorgt dafür, dass Produkte in das Modell kommen.
- Queue, dieses Atom ist eine Warteschlange für Kunden oder Produkte.
- Server, dieses Atom dient als Maschine oder Schalter. Atome erfahren hier eine Modifizierung und bleiben für eine bestimmte Zeit (die Bearbeitungszeit) in diesem Atom.
- Sink, dieses Atom sorgt dafür, dass Kunden oder Produkte das Modell wieder verlassen können.



Abbildung 4-1: Speed Buttons: Source, Queue, Server und Sink

Fallbeispiel 1

Ein Postamt in Amsterdam wird im Schnitt von 20 Kunden pro Stunde besucht, und der Mitarbeiter des Postamts braucht durchschnittlich zwei Minuten, um einen Kunden zu bedienen. Natürlich wird das Postamt in einigen Zeitabschnitten häufiger frequentiert als in anderen. Auch die zwei Minuten, die der Mitarbeiter des Postamts benötigt, variieren. Ein Kunde, der nur Briefmarken kaufen möchte, braucht schließlich weniger Zeit als ein Kunde, der ein Konto eröffnen will. Die Kunden werden in der Reihenfolge ihres Eintreffens bedient.

Ein paar Kunden haben sich über die Warteschlangen beschwert, und weil der Manager des Postamts dem Service gegenüber seinen Kunden große Bedeutung beimisst, möchte er dies untersuchen lassen.

Fragen und Aufgaben

1. Wie hoch ist die Auslastung des Mitarbeiters? Was bedeutet dies?
2. Kann auf der Grundlage dieser Informationen eine Schätzung der durchschnittlichen Warteschlange vorgenommen werden?
3. An welchen Ergebnissen einer Studie wird der Manager interessiert sein?

Simulation ist eine Möglichkeit, Einblick in die durchschnittliche Warteschlange zu erhalten. Weil dies das erste Modell ist, das wir generieren werden, wird es Schritt für Schritt erläutert.

Wir nutzen hierfür die 4 Atome, die zu Beginn dieses Abschnitts erörtert wurden, und platzieren diese in der oben dargelegten Reihenfolge in unserem Modell. Zunächst wird also die Source platziert, anschließend die Queue, dann der Server und schließlich die Sink. Die Platzierung der Atome erfolgt durch Anklicken der Speed Buttons (siehe Abbildung 4-1) oder durch das Ziehen der genannten Atome aus der Library.

Wenn dieser Schritt getan ist, entsteht Abbildung 4-2:

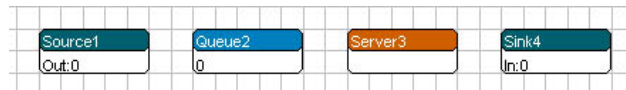


Abbildung 4-2: das erste Modell

Bei irrtümlichen Falschplatzierungen reicht es, ein Atom anzuklicken und es anschließend mit der Löschtaste zu entfernen.

Um zu überprüfen, ob die Kunden sich in der richtigen Reihenfolge durch das Modell bewegen, starten wir die Simulation. Dies tun wir mit Hilfe des Run Control Fensters. Wenn dieses Fenster noch nicht geöffnet ist, muss dies zunächst über die Option Simulate im Hauptmenü getan werden. Achten Sie bitte darauf, dass im Run Control Fenster die Einstellung Slide Control ausgewählt ist (siehe Abbildung 4-3). Hierdurch ist es möglich, mit Hilfe des Schiebereglers unten im Run Control Fenster die Geschwindigkeit der Simulation anzupassen.

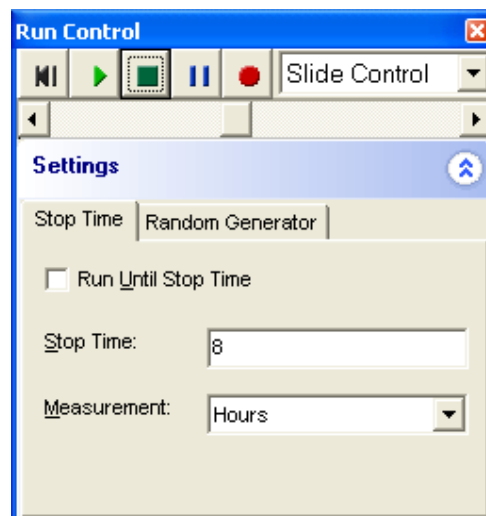


Abbildung 4-3: wählen Sie Slide Control aus

Bevor die Simulation gestartet werden kann, muss zunächst der Reset Button gedrückt werden. Dies ist der Button ganz links im Run Control Fenster. Nachdem der Reset Button gedrückt wurde, erscheint ein blauer Kreis neben dem Source Atom. Dies ist ein Product Atom. Wenn Sie jetzt auf den Startbutton drücken (grünes Dreieck), werden sich Produkte, in diesem Fall Kunden, durch die anderen Atome, hier das Postamt, bewegen. Passen Sie erforderlichenfalls die Geschwindigkeit an!

In der Regel wird in diesem Beispiel keine Warteschlange vor dem Schalter (Server) erscheinen. Das blaue Produkt ist also nur im Schalter (Server) sichtbar. Daneben wird im Server ein Prozentsatz angezeigt. Dieser Prozentsatz stellt die Auslastung des Servers dar. Wenn dieser Pro-

zentsatz nicht sichtbar ist, kann es erforderlich sein, in die Ansicht herein- oder herauszuzoomen. Dies ist möglich, wenn Sie gleichzeitig die linke und die rechte Maustaste drücken und die Maus vorwärts oder rückwärts bewegen. Die Bewegungen des Product Atoms lassen sich besser verfolgen, wenn Sie 2D Visual Trace – ein Untermenü der Option Display – aktivieren.

Wir haben nun die Atome in der richtigen Reihenfolge platziert, allerdings noch nicht spezifiziert, wie viele Kunden pro Stunde eintreffen und wie viel Zeit der Mitarbeiter benötigt, um einen Kunden zu bedienen. Bevor wir diese Zeiten in Enterprise Dynamics eingeben können, müssen wir berücksichtigen, dass alle Zeiten in Enterprise Dynamics in Sekunden eingegeben werden. Wenn der Mitarbeiter zwei Minuten benötigt, um einen Kunden zu bedienen, geben wir dies in Enterprise Dynamics als 120 Sekunden ein. Daneben ist es nur möglich, zu spezifizieren, wie viel Zeit zwischen dem Eintreffen von jeweils zwei Kunden liegt, die so genannte Zwischenankunftszeit (ZAZ).

Zunächst ändern wir die Einstellungen der Source, so dass pro Stunde durchschnittlich zwanzig neue Kunden im Postamt eintreffen. Durch Doppelklicken des Source Atoms öffnet sich ein Eingabefenster (siehe Abbildung 4-4). Im Feld Inter-arrival time kann die Zwischenankunftszeit eingegeben werden. Die Unsicherheit im Ankunftsprozess kann durch Nutzung einer Wahrscheinlichkeitsverteilung abgefangen werden. In diesem Beispiel verwenden wir die negativ-exponentielle Verteilung, eine Wahrscheinlichkeitsverteilung, die häufig für chaotische Ankunfts- oder Bedienungsprozesse genutzt wird. In Enterprise Dynamics muss dafür der folgende 4DScript Code verwendet werden: `negexp(e1)`. Hierbei steht `e1` für den durchschnittlichen Wert oder die Erwartung der exponentiellen Verteilung. Geben Sie nun als Zwischenankunftszeit 180 ein und bestätigen Sie mit OK:

The image shows a software window titled "ED Source - Source1". It has two tabs: "General" and "Visualization". The "General" tab is selected. Inside the "General" tab, there is a section for "Atom name" with a text box containing "Source1". Below this is a "Settings" section with four dropdown menus: "Inter-arrival time [s]" set to "4DS NegExp(180)", "Time till first product [s]" set to "4DS NegExp(180)", "Number of products" set to "1. Unlimited", and "Send to" set to "1". Below the settings is a "Triggers" section with two dropdown menus: "Trigger on creation" set to "10. Do Nothing" and "Trigger on exit" set to "4DS 0". At the bottom of the window are four buttons: "Help", "Ok", "Cancel", and "Apply".

Abbildung 4-4: Eingabefenster Source

Wichtig: Rechts von den Eingabefeldern steht häufig ein nach unten zeigendes Dreieck. Ein Doppelklick ergibt eine Liste mit verfügbaren und vordefinierten Optionen. Schauen Sie zuerst hier nach, ob eine für Ihr Problem brauchbare Option enthalten ist! Eine Erläuterung der Optionen finden Sie in Anlage 2. Hierin sind die wichtigsten Atome und ihre Eingabefelder vollständig beschrieben. Der Text 4DS bedeutet, dass hier 4DScript Befehle (die Programmiersprache von ED) eingegeben werden können.

Anschließend geben wir auch die Zeit ein, die ein Mitarbeiter benötigt, um einen Kunden zu bedienen. Auch hier verwenden wir wieder die negativ-exponentielle Verteilung. Doppelklicken Sie nun das Server Atom, sodass die Bearbeitungszeit angepasst werden kann (siehe Abbildung 4-5). Passen Sie die Cycletime nun so an, dass der Mitarbeiter durchschnittlich zwei Minuten für einen Kunden benötigt.

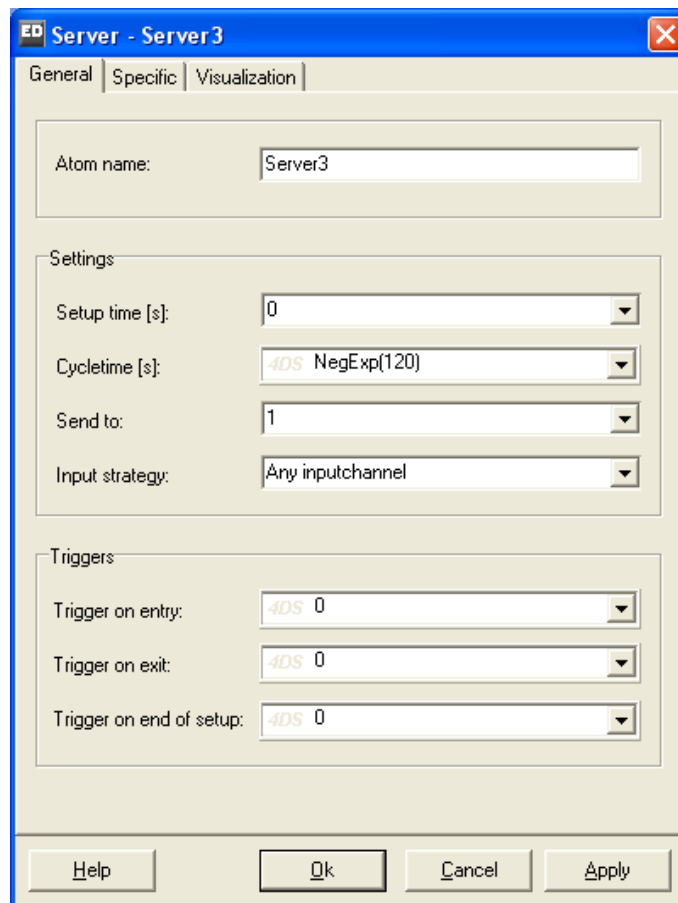


Abbildung 4-5: Eingabefenster Server

Resetten und starten Sie mit der Run Control die Simulation erneut. Da wir die Zeit zwischen den beiden Ankunftsmomenten erhöht haben, empfiehlt es sich, die Geschwindigkeit der Simulation zu steigern. Die Geschwindigkeit der Simulation wird weiter gesteigert, wenn 2D Visual Trace wieder deaktiviert wird. Durch hinreichendes Hereinzoomen ist sichtbar, wie viele Kunden in der Schlange warten.

Es ist möglich, das blaue Produkt, das jetzt einen Kunden symbolisiert, durch eine andere Abbildung zu ersetzen. Doppelklicken Sie hierfür den blauen Kreis links von der Source. Beim Eingabefeld 2D Icon doppelklicken Sie erneut den blauen Kreis. Es erscheint der Resource Manager mit den Icons, die vom Anwender ausgewählt werden können. Klicken Sie auf das gewünschte Icon und auf OK im Resource Manager: der blaue Kreis im Produkt Fenster wurde durch das neue Icon ersetzt.

Auch das Icon, das Enterprise Dynamics in der 3D Darstellung verwendet, kann angepasst werden. Nutzen Sie hierfür die Option 3D Icon und treffen Sie eine Auswahl aus der Liste für Personen. Betrachten Sie das Modell jetzt in 2D und in 3D. Zum Öffnen der 3D Darstellung wählen Sie die Hauptmenüoption Display und anschließend eine der Möglichkeiten der 3D Darstellung. Durch Bewegen der Maus und Klicken der Maustasten kann durch das 3D Modell navigiert werden.

Tipp: Es sind viel mehr Icons verfügbar. Versuchen Sie, über File|Import die Standardliste um ein bestehendes Icon zu ergänzen!

In dieser Simulationsstudie sorgt die Auswahl eines anderen Icons lediglich für eine schönere Simulation. Wenn jedoch zwischen verschiedenen Arten von Kunden unterschieden werden müsste, kann die Verwendung mehrerer Icons für mehr Deutlichkeit sorgen. Eine Kundengruppe kann dann zum Beispiel ein grünes und die andere ein blaues Icon erhalten.

Das Modell kann nun gespeichert werden. Zur Kontrolle ist das Modell dieses Postamts bei den Tutorial-Modellen unter der Bezeichnung Postoffice1.mod zu finden.

4.2 Die Kanäle

Fallbeispiel 1 (Fortsetzung)

In den umliegenden Stadtteilen werden 2 Postämter geschlossen. Der Leiter hofft, dass sein Postamt viele neuen Kunden begrüßen wird und rechnet sogar mit einer Verdopplung der Kunden. Es werden also 40 Kunden pro Stunde sein Postamt besuchen. Um herauszufinden, ob sein Postamt mit dem einen bestehenden Schalter diesen Zustrom verarbeiten kann, beschließt er, erneut eine Simulationsstudie ausführen zu lassen.

4. Ist es sinnvoll, in dieser Situation eine Simulation vorzunehmen? Berücksichtigen Sie bei der Beantwortung dieser Frage die zu erwartende Auslastung und prüfen Sie die eigene Annahme, indem Sie das Modell Postoffice1.mod an die neue Situation anpassen und eine Simulation ausführen.

Es zeigt sich, dass ein Schalter nicht ausreichen wird, den Zustrom zu verarbeiten. Der Leiter möchte zur Sicherheit sogar drei Schalter öffnen. Erneut möchte er gern einen Einblick in die Auswirkungen dieser Maßnahme auf die Warteschlange erhalten.

Um zu beginnen, platzieren wir die beiden Schalter unter den bereits bestehenden Schalter (wählen Sie das richtige Atom zur Modellierung eines Schalters). Anschließend müssen die Einstellungen für die Cycletime der Schalter geändert werden, sodass sie mit dem ersten Schalter übereinstimmen. Nutzen Sie hierbei gegebenenfalls Abschnitt 4.1. Das Modell müsste nun Abbildung 4-6 entsprechen.

Tipp: Atome können ganz einfach – mit allen ausgefüllten Feldern und Einstellungen – kopiert werden! Klicken Sie dazu auf das zu kopierende Atom und drücken Sie anschließend F6. Wählen Sie für das Kopieren mehrerer Atome mit einem Rechtsklick der Maus einen Bereich und drücken Sie anschließend wieder F6 oder Ctrl-V. Achtung: mit Ctrl-V wird die Kopie meist genau unter dem Original liegen, verschieben Sie deshalb das Original oder die Kopie.

Die oben beschriebenen Verfahren in Kombination mit dem Cursortasten eignen sich für das gleichzeitige Verschieben mehrerer Atome.

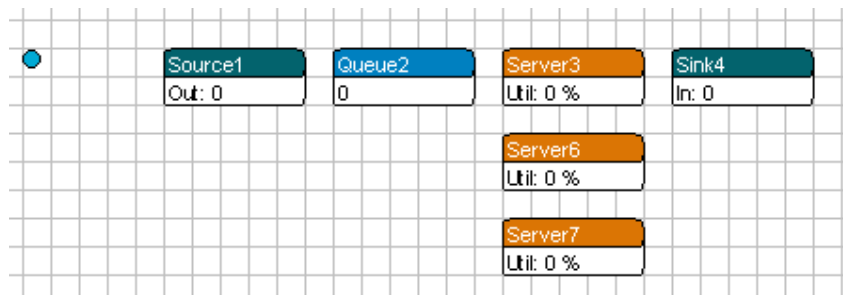


Abbildung 4-6: Postamt mit drei Schaltern

Wenn die Simulation nun gestartet wird, bewegen sich die Kunden allerdings noch immer nur zum obersten Schalter. Um dies anzupassen, muss berücksichtigt werden, wie Enterprise Dynamics die Kunden weiterschickt.

Wählen Sie im Menü View des Model Layout Fensters die Option Channels | Enabled (siehe Abbildung 4-7). Hierdurch werden die so genannten Kanäle (Channels) der Atome sichtbar. Diese Kanäle dienen in ED zur Modellierung der Material- und Informationsströme und werden deshalb an dieser Stelle ausführlich erörtert.

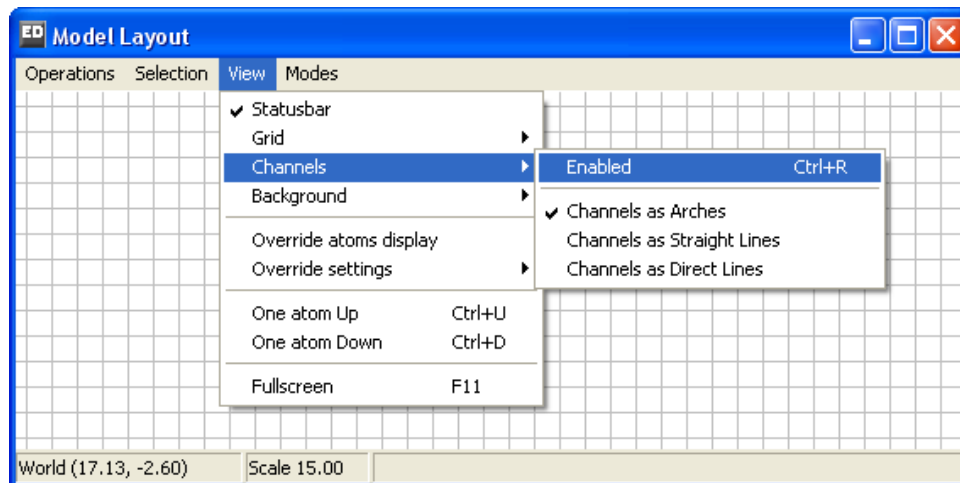


Abbildung 4-7: Kanäle aktivieren

Wenn die Linien zwischen zwei Kanälen nicht als Bögen dargestellt werden, sondern als eckige Linien, empfiehlt es sich, die Option Channels as Arches im Menü View | Channels zu aktivieren (siehe wieder Abbildung 4-7).

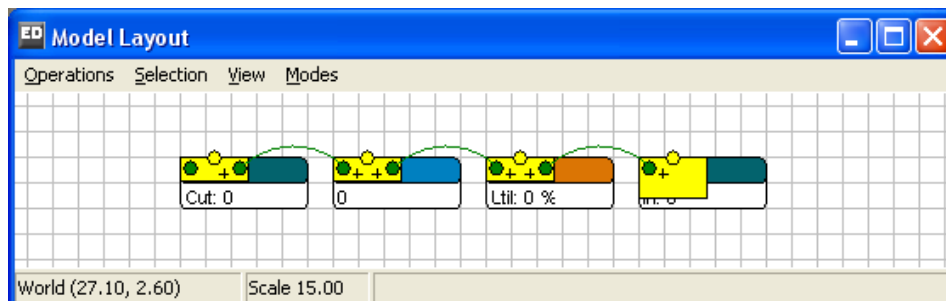


Abbildung 4-8: das Postamt Modell mit den aktivierten Kanälen

Sobald die Kanäle aktiviert wurden, erhalten alle Atome ein Kästchen, auf dem die Ein- und Ausgangskanäle des betreffenden Atoms angezeigt werden (siehe Abbildung 4-8).

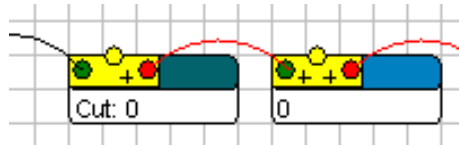


Abbildung 4-9: Verbundene Kanäle

Im Kästchen sind einige Kreise zu sehen. Der linke Kreis ist ein Eingangskanal, der rechte ein Ausgangskanal. Ein Kanal kann *geöffnet* (dargestellt in Grün) oder *geschlossen* (dargestellt in Rot) sein. Wenn sowohl der Ein- als auch der Ausgangskanal geöffnet sind, ist die Verbindung zwischen dem Ein- und Ausgangskanal *bereit* (dargestellt in Grün), andernfalls ist die Verbindung *nicht bereit* (dargestellt in Rot).

Der mittlere Kreis (oben auf dem gelben Kästchen) dient zum Versenden und Empfangen von Informationen, in diesem Tutorial werden wir diesen Kanal als „zentralen Kanal“ (central channel) bezeichnen. Dieser zentrale Kanal wird dazu genutzt, Informationen über ein Atom zu registrieren.

Auf einem Atom ist immer nur ein zentraler Kanal verfügbar, es ist allerdings möglich, mehrere Atome mit demselben zentralen Kanal zu verbinden.

Produkte gelangen über einen Eingangskanal in ein Atom und verlassen es über einen Ausgangskanal. Ein Eingangskanal muss immer mit einem Ausgangskanal oder einem Mittelkanal verbunden sein. Ein Ausgangskanal muss immer mit einem Eingangskanal oder einem Mittelkanal verbunden sein.

Jeder Eingangs- oder Ausgangskanal kann nur mit einem anderen Kanal verbunden sein.

Durch Klicken des + Zeichens neben einem Kreis wird die Zahl der Ein- oder Ausgangskanäle erhöht, und durch Rechtsklick auf einen Kanal erhält man eine Übersicht aller Verbindungen des jeweiligen Atoms.

Wichtig:

Obwohl mit Kanälen sehr viel geregelt werden kann, besteht das Wesen von Kanälen darin, auf einem Atom (sprich: Warteschlange, Schalter) alle möglichen folgenden Orte von Produkten, die sich in diesem Atom befinden, aufzuzeigen. Damit sorgen die Kanäle für ein (grobes) Routing von Produkten durch das Modell!

In Abbildung 4-10 ist ein Atom mit 4 Eingangskanälen, 6 Ausgangskanälen und dem zentralen Kanal dargestellt.

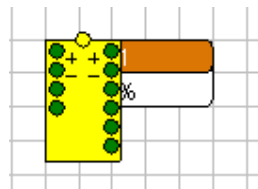


Abbildung 4-10: Atom mit 4 Eingangskanälen und 6 Ausgangskanälen

Indem mit dem Mauszeiger eine Linie zwischen einem Ausgangskanal und einem Eingangskanal gezogen wird, werden zwei Atome miteinander verbunden. Durch das Ziehen einer Linie zwischen einem Ein- oder Ausgangskanal und dem zentralen Kanal desselben Atoms wird eine Verbindung gelöscht.

Verbinden Sie die Kanäle im Modell in Enterprise Dynamics nun so, dass die Kunden zu einem der drei Schalter gehen und sich anschließend von einem Schalter aus zum Ausgang bewegen. Das Ergebnis sollte aussehen wie in Abbildung 4-11.

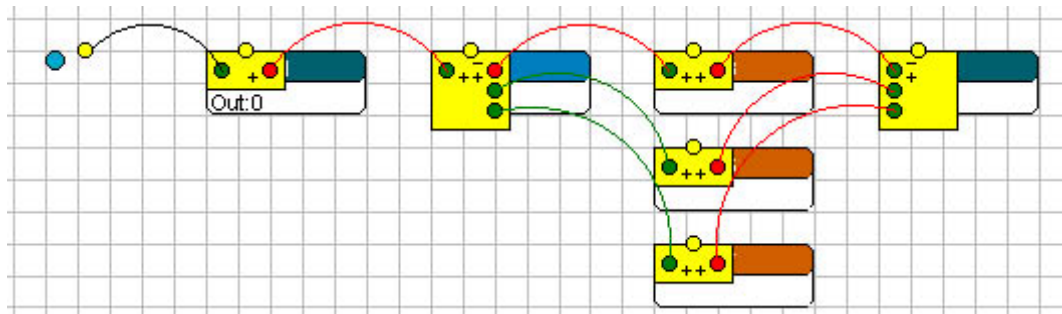


Abbildung 4-11: Postamt mit verbundenen Kanälen

Starten Sie die Simulation neu (Reset bitte nicht vergessen!) und kontrollieren Sie, ob die Kunden jetzt zu den richtigen Schaltern gehen.

Der Grund dafür, dass die Kunden noch immer nur den obersten Schalter wählen, besteht darin, dass in Enterprise Dynamics noch definiert ist, dass Kunden (Atome) standardmäßig über Ausgangskanal 1 weitergeschickt werden sollen. Im Eingabefenster, das beim Doppelklicken eines Atoms sichtbar wird, lässt sich einstellen, über welchen Kanal die Produkte die Warteschlange verlassen sollen. Durch Doppelklicken der Queue können wir die gewünschten Einstellungen für unser Modell vornehmen. Wir müssen hierfür den Wert im Feld Send To anpassen (siehe Abbildung 4-12): wenn wir möchten, dass jeder offene Kanal gewählt werden kann, sind die Optionen 2, 3 und 4 relevant.

Begründen Sie bitte, warum bei Option 2 bzw. Option 3 der erste bzw. letzte Server den höchsten Auslastungsgrad erhält, während die Auslastungsgrade der Server bei Option 4 alle mehr oder weniger identisch sind!

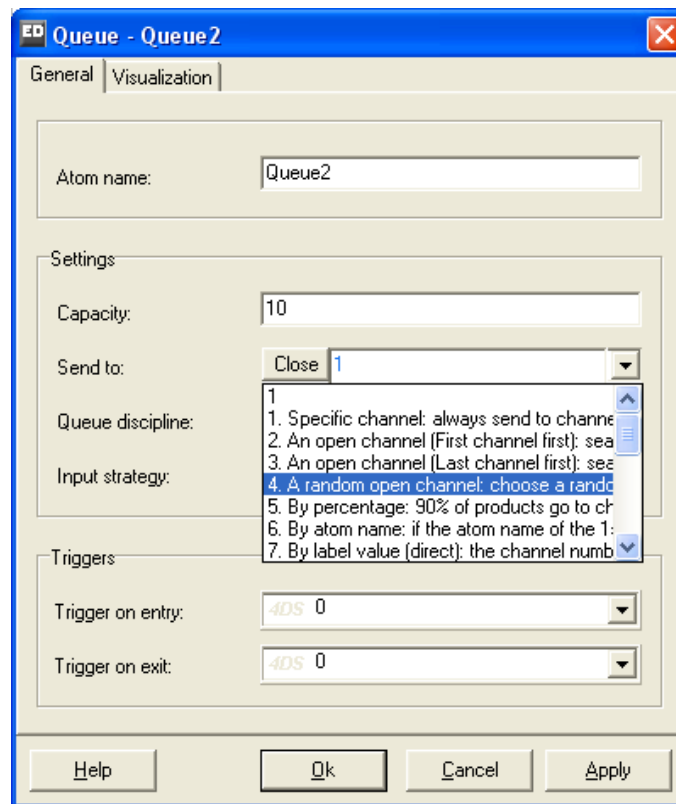


Abbildung 4-12: Send to bei der Queue

Beim Send To Eingabefeld können Sie eine Zahl eingeben, mit 4DScript eigenen Programmcode schreiben, welcher die Ausgangszahl festsetzt, oder vordefinierten Code nutzen. Letztgenannte Möglichkeit ist vor allem für Anwender auf Anfängerniveau die bequemste Methode. Im Drop-down-Menü des Eingabefelds erscheint eine Liste der zur Verfügung stehenden Optionen. Nachdem die Wahl für eine bestimmte Regel getroffen wurde, können die blauen Textstellen angepasst werden. Selbstverständlich besteht diese Option nur, wenn Anpassungen möglich sind.

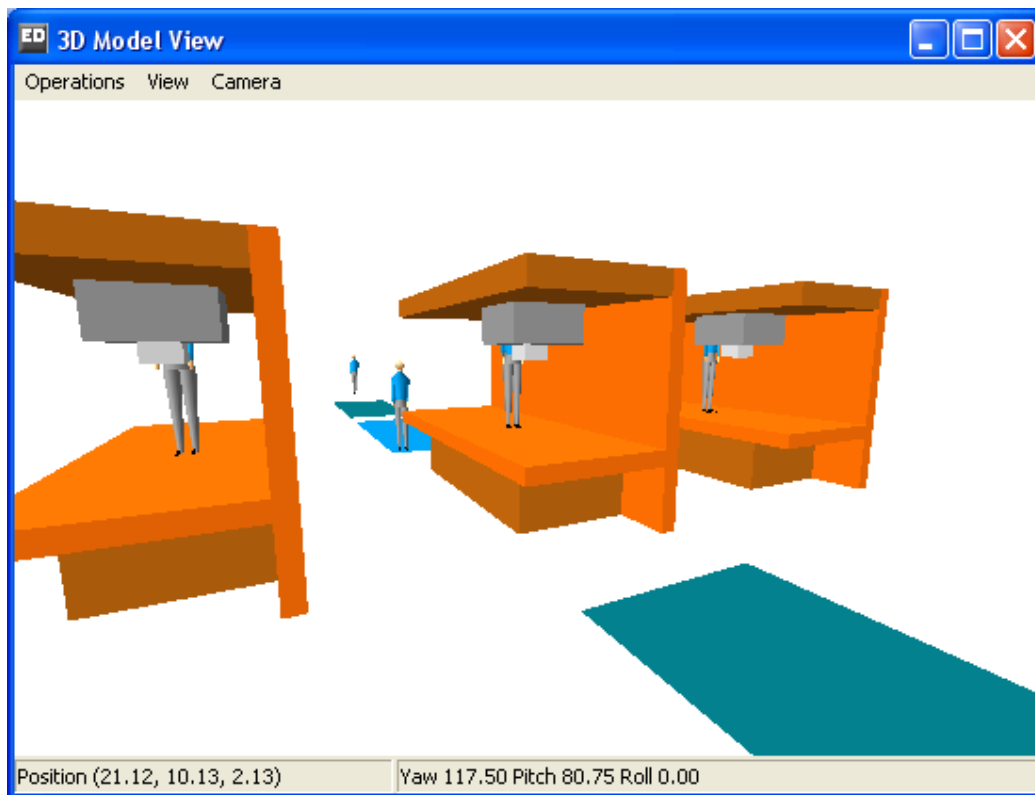


Abbildung 4-13: Das 3D-Modell mit drei Schaltern, modelliert als Maschinen

Wählen Sie nun bitte selbst eine Strategie, bei der die Auslastung der drei Schalter ungefähr identisch ist. Wenn die Auslastungsgrade der Schalter nicht direkt sichtbar sind, muss herein- oder herausgezoomt werden (mit Hilfe beider Maustasten).

5. Ist die Aufstellung mit drei Schaltern ausreichend?

Das Modell kann nun gespeichert werden. Zur Kontrolle ist das Modell dieses Postamts bei den Tutorial-Modellen unter der Bezeichnung Postoffice2.mod zu finden.

5 ERGEBNISSE ANSCHAUEN

Im vorigen Abschnitt wurde anhand eines umfassenden Stufenplans ein Postamt simuliert. Dabei haben Sie gelernt, wie Atome im Modell platziert werden und wie die Kanäle verbunden werden. Daneben wurde erläutert, wie dafür gesorgt werden kann, dass die Atome zu den richtigen Zeitpunkten in das Modell kommen und für die richtige Zeitspanne bei den Maschinen (oder Schaltern) bleiben. Alle Grundprinzipien für die Generierung eines Modells wurden folglich behandelt.

In diesem Abschnitt generieren wir ein komplexeres Modell und schauen vor allem auf Möglichkeiten, die Ergebnisse der Simulationsstudie zu verfolgen und zu messen. Zunächst folgt wieder ein (einfaches) Fallbeispiel und, nach der Generierung eines dazugehörigen Modells, eine Erläuterung der verschiedenen Methoden der Ergebnisnahme. Anschließend üben wir deren Anwendung im Modell.

Fallbeispiel 2 Die Zimmerei

Allgemeiner Kontext:

Eine Zimmerei hat meist eine eigene Entwurfsabteilung und, daraus erwachsend, eine komplexe Steuerung, weil jede Bestellung im Grunde ein separates Produkt ist, dessen Bearbeitungszeiten und einzelne Produktionsschritte möglichst gut eingeschätzt werden müssen. Aus der Produktvielfalt ergeben sich große Zwischenvorräte, lange Durchlaufzeiten und wechselnde Engpässe.

Die Geschäftsleitung einer Zimmerei, die hauptsächlich Fenster und Rahmen herstellt, möchte einen besseren Einblick in die Problembereiche, Produktionsmengen und Durchlaufzeiten erhalten. Deshalb wird beschlossen, eine Simulationsstudie durchführen zu lassen. Der Einfachheit halber wird in diesem Fallbeispiel lediglich die Produktion von Fenstern berücksichtigt.

Der Produktionsprozess für die Fenster lässt sich in eine Reihe von Schritten unterteilen. Zunächst treffen lange Balken ein, die mit der Querkreissäge in kurze Stücke zerteilt werden. Hierbei werden aus einem langen Balken zehn kurze Teilstücke gesägt. Diese kleinen Balken erhalten anschließend auf der Fräsmaschine das gewünschte Profil. Nach dem Fräsen gelangen die Balken zur Verschlussbank. Hier werden jeweils 4 Balken in einen Rahmen gelegt und schnell verleimt. Sowohl von der Fräsmaschine als auch von der Verschlussbank sind zwei identische Exemplare verfügbar, die parallel zueinander in den Produktionsprozess integriert sind. Die Balken werden also nur einmal gefräst und verleimt. Zwischen allen Produktionsschritten liegen jeweils Vorratsstellen mit hinreichender Platzkapazität. Es sind genügend Balken vorhanden, um die Querkreissäge kontinuierlich laufen zu lassen. Um den Holzvorrat im System überblicken zu können, gilt, dass die Zwischenvorräte je Bearbeitungsschritt nicht mehr als 100 Balken umfassen dürfen.

Das Sägen eines Balkens auf der Querkreissäge dauert gleichverteilt zwischen zwei und drei Minuten, die Bearbeitungszeit eines Balkens auf der Fräsmaschine ist normal verteilt, mit einem Durchschnitt von 36 Sekunden und einer Standardabweichung von 2 Sekunden. Die Verschlussbank benötigt genau zwei Minuten für das Zusammenleimen eines Fensters. Die Produktion läuft ohne Unterbrechung von 9.00 bis 17.00 Uhr. Die Produkte, die am Ende des Tages noch nicht fertig gestellt sind, bleiben bis zum nächsten Tag liegen.

Analyse im Vorfeld

1. Erstellen Sie eine Skizze des Prozesses und geben Sie bei jedem Teilprozess die Kapazität pro Stunde für Balken oder Fenster an.
2. Wie viele Fenster verlassen voraussichtlich pro Tag die Fabrik?

Modellieren Sie nun die Zimmerei in Enterprise Dynamics. Die Wahrscheinlichkeitsverteilungen sind im Pulldown-Menü des Parameterfeldes Cyclic times zu finden; achten Sie darauf, dass für die Normalverteilung der Ausdruck $\max(0, \text{normal}(\ , \))$ verwendet wird, um zu verhindern, dass negative Bearbeitungszeiten entstehen. Nutzen Sie die Batch Rule auf dem Server für die 1:10 Operation zur Teilung von Balken zu kurzen Stücken und die 4:1 Operation bei der Fertigung von Fenstern aus kurzen Balken. Ändern Sie die Servernamen, sodass das Modell leichter gelesen werden kann.

Wenn alles richtig gelaufen ist, entsteht ein Modell wie in Abbildung 5-1.

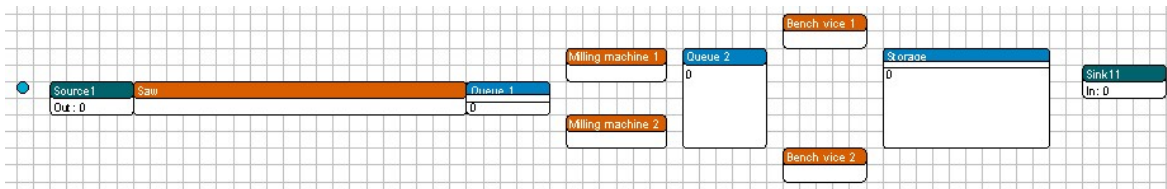


Abbildung 5-1: Layout der Zimmerei

Das Produkt Atom (der blaue Kreis) lässt sich in einen braunen „Balken“ ändern: doppelklicken Sie das Produkt und ändern Sie das Eingabefenster wie unten dargestellt.

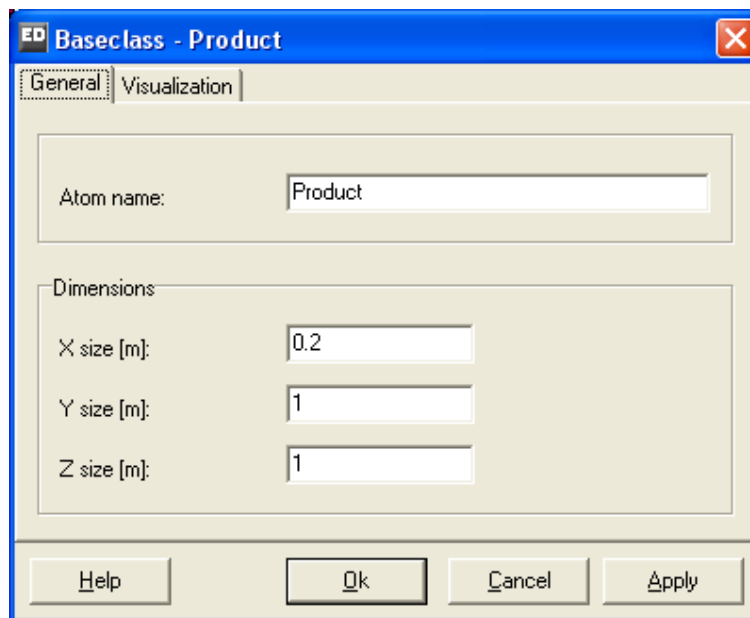
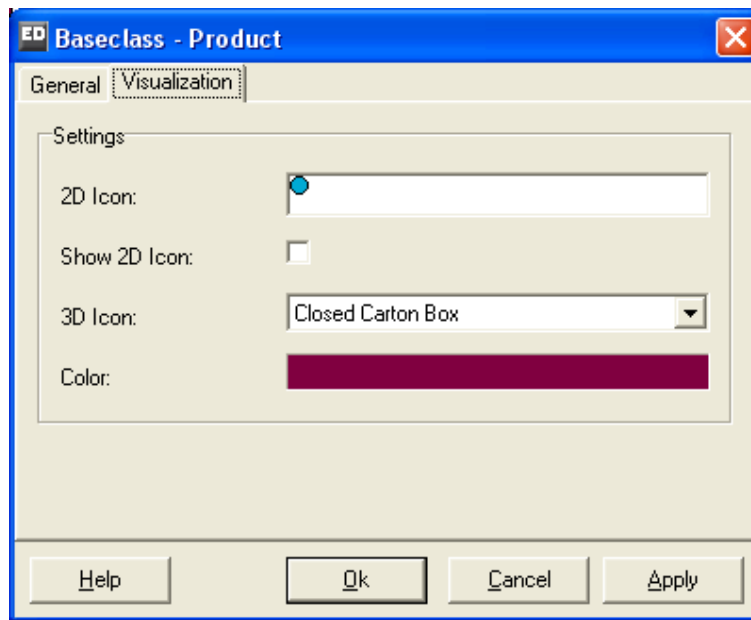


Abbildung 5-2: Eingabefenster des Produkts



Bestätigen Sie die Änderungen gegenüber dem ursprünglichen Eingabefenster. Was ist nun geschehen?

Das 2D Icon mit dem blauen Kreis ist immer noch da, aber durch das Deaktivieren der Option *Show 2D Icon* ist lediglich die braune Hintergrundfarbe sichtbar. Diese hat eine Länge von 0.2 m und eine Breite von 1 Meter wie bei einem stehenden Balken. In 3D spielt die Höhe (Size Z) von 0.2 Metern noch keine Rolle. Beachten Sie, dass das Produkticon in 2D sich von der 3D-Ansicht unterscheiden kann. Wählen Sie in 3D zum Beispiel die Palette oder ein anderes Icon!

Das Raster, das jeweils hinter den Atomen sichtbar ist, fungiert als Koordinatenfläche, wobei jedes Fach eine Abmessung von 1 x 1 Meter hat. Das sichtbare schwarze Fach entspricht den Koordinaten (0,0). Für dieses abstrakte Modell einer Zimmerei spielt der physische Raum keine Rolle, aber bei anderen Anwendungsbereichen, wie zum Beispiel Lager mit Fußwegen oder Staplern, die eine bestimmte Entfernung zurücklegen müssen, ist dies durchaus von Bedeutung!

Speichern Sie das eigene Modell. Bei Unsicherheit bezüglich des eigenen Modells kann für die nächsten Abschnitte auch das mitgelieferte Modell *Timber1.mod* verwendet werden.

5.1 Methoden zur Messung der Ergebnisse

Mit einem funktionsfähigen Modell können wir nun mit der Messung der Ergebnisse beginnen. Enterprise Dynamics bietet einige Methoden zur Messung dieser Ergebnisse.

1. Informationen, die auf den Atomen dargestellt sind. So wird bei jeder Queue angegeben, wie viele Produkte sich in der Warteschlange befinden, auf einem Server steht die Auslastung und auf einer Source oder Sink steht, wie viele Produkte das Modell erreicht bzw. verlassen haben. Diese Informationen lassen sich vor allem dazu nutzen, während einer Simulation zu prüfen, ob das Modell logisch richtig funktioniert.
2. Ein Monitor. Dieser Monitor gibt grafische Informationen über ein bestimmtes Atom. Der Anwender kann einstellen, wie viele Informationen angezeigt werden sollen. Auch diese In-

formationen lassen sich vor allem dazu nutzen, während einer Simulation zu prüfen, ob das Modell logisch richtig funktioniert.

3. Übersichtsberichte und Grafiken aus dem Results Menü. Hierin erscheinen (zwischenzeitliche) Ergebnisse eines laufenden Simulationsdurchlaufs. Es ist vor allem eine praktische Methode, um schnell einen Überblick über den Zustand des Systems und eine schnelle Kontrolle bezüglich der Parametereinstellungen zu erhalten.
4. Ein Experiment. Diese Methode unterscheidet sich wesentlich von den vorigen 3 und wird für die eigentliche Studie genutzt. Zuvor wird eingestellt, wie lange ein Messintervall dauert und wie oft dieser Zeitraum simuliert werden soll, z.B. 10 mal ein halbes Jahr. Außerdem muss bei einem Experiment angegeben werden, welche Größe gemessen werden soll. Das Ergebnis eines Experiments liegt in Form eines Vertrauens- oder Konfidenzintervalles vor.

5.2 Das Messen der Ergebnisse

In Abschnitt 5.1 wurden vier Methoden behandelt, mit denen die Ergebnisse einer Simulationsstudie angezeigt werden können. In diesem Abschnitt werden wir diese vier Methoden bei der Zimmerei einsetzen. Sorgen Sie dafür, dass das Modell der Zimmerei in Enterprise Dynamics geöffnet ist.

Zu 1. Informationen, die auf dem Atom dargestellt werden

Sorgen Sie dafür, dass die Uhr von Enterprise Dynamics auf dem Bildschirm sichtbar ist (Untermenü von Simulate). Starten Sie anschließend die Simulation und schauen Sie, wie viele Produkte in acht Stunden die Fabrik verlassen. Mit Hilfe der Option *Simulate | Set StopTime* und anschließend *Simulate | Reset + Run until Stoptime* kann Enterprise Dynamics so eingestellt werden, dass die Simulation nach genau 8 Stunden stoppt.

3. Führen Sie dies einige Male mit *Simulate | Reset + Run until stoptime* aus und notieren Sie die Zahl der produzierten Fenster. Entspricht diese der zuvor ausgeführten Analyse? Wo liegt der Engpass bei diesem Prozess?

Zu 2. Ein Monitor

Die Nutzung eines Monitors ist recht einfach. Sorgen Sie dafür, dass der Library Tree geöffnet ist (nicht der Model Tree) und suchen Sie das Atom Monitor in der Gruppe Results. Ziehen Sie das Monitor Atom in das Modell. Wenn die Kanäle sichtbar gemacht sind, müsste das Atom wie in Abbildung 5-3 aussehen.

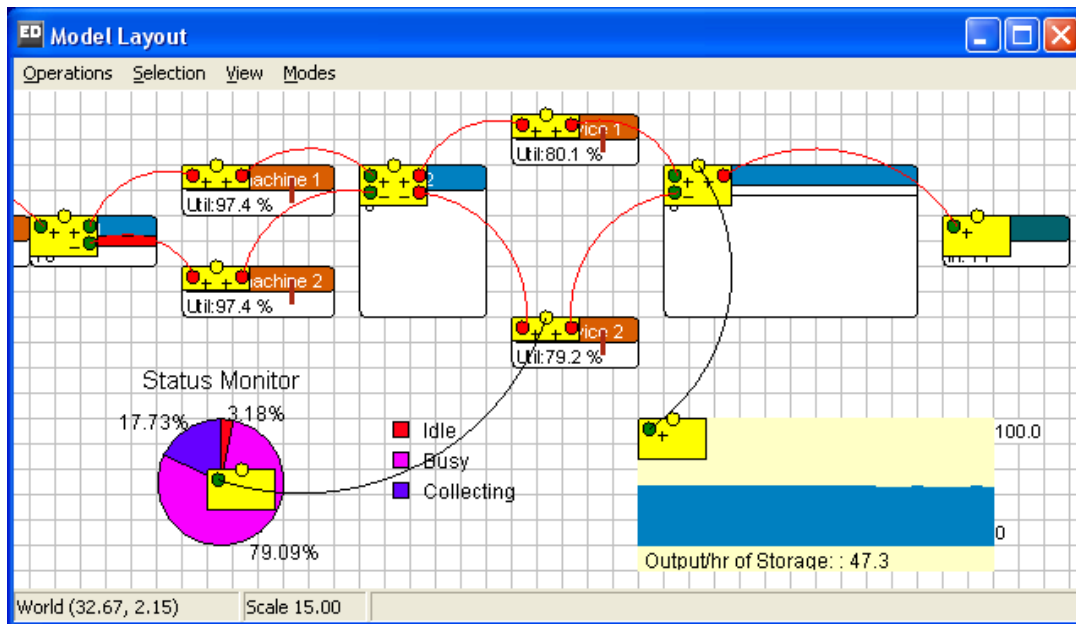


Abbildung 5-3: Modellteil mit einem (bereits verbundenen) Monitor Atom

Der erste Schritt besteht in der Verbindung des Monitors mit dem Atom, das betrachtet werden soll. In diesem Fallbeispiel möchten wir wissen, wie viele Fenster pro Stunde die Fabrik verlassen. Verbinden Sie den Eingangskanal des Monitors mit dem Mittelkanal (dem Informationskanal!) der letzten Queue. Es ist auch möglich, den Monitor doppelzuklicken und anschließend das Atom auszuwählen, das betrachtet werden soll.

Im zweiten Schritt wird definiert, welche Größe betrachtet werden soll. Doppelklicken Sie hierfür erneut das Monitor Atom und wählen Sie die Option Monitor Variable. Nun sollte eine Liste wie in Abbildung 5-4 erscheinen.

In diesem Fallbeispiel benötigen wir die Option „Output per hour“, also den Durchsatz pro Stunde.

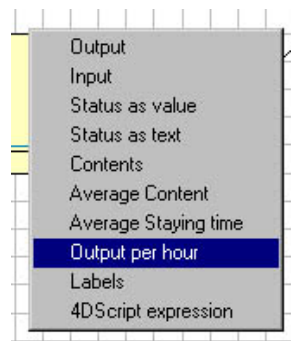


Abbildung 5-4: Monitor Variable

Wenn die gewünschte Einflussgröße nicht in der Liste enthalten ist, ist es erforderlich, sie selbst in 4DScript Code zu programmieren. Verwenden Sie hierfür die Option 4DScript expression. Für Anwender auf Anfängerniveau ist dies jedoch nicht zu empfehlen.

Im dritten Schritt können noch einige Visualisierungseinstellungen für den Monitor vorgenommen werden. Wählen Sie nach dem Doppelklicken des Monitors die Option Set Preferences. Das Fenster aus Abbildung 5-5 erscheint dann:

Monitor - Monitor11

General | Visualization

Settings

Enter subscript: Output/hr of Storage

Refresh rate [s]: 60

Number of decimals: 1

Show result as text: ☐

Preferences

Graph type: Bargraph

Variable to monitor: Output per hour

4D Script expression: 4DS

Label to monitor:

Show dynamic icon: ☒

Auto adjust: ☐

Show Y-axis: ☒

Maximum value: 100

Histogram start value: 0

Histogram nr. of classes: 30

Histogram class size: 1

Miscellaneous

Select Atom

Help Ok Cancel Apply

Abbildung 5-5: Monitor preferences

Die *Sample rate* ist standardmäßig auf 5 Sekunden eingestellt. Stellen Sie diese auf 60: nun wird pro Minute gemessen. Testen Sie *Auto Adjust*, *Maximum value* und *Show Y-Achse*, um ein Gefühl für die verschiedenen Einstellungen zu bekommen. Starten Sie nun die Simulation.

4. Wie viele Fenster verlassen durchschnittlich pro Stunde die Fabrik?

Zu 3. Reports und Grafiken

Eine dritte Methode zur Messung von Ergebnissen ist die Nutzung der Option Reports und Grafiken aus dem Results Menü.

Zunächst wird die Option Reports besprochen. Hiermit können Sie sich einen Überblick über den Zustand des Modells verschaffen, wobei eine Unterteilung in folgende Punkte vorgenommen wird:

- Current content: die Zahl der Produkte, die zum Zeitpunkt der Erstellung des Reports in einem Atom vorhanden sind.
- Average content: die durchschnittliche Zahl der Produkte, die im Atom vorhanden waren.
- Input: die Zahl der Produkte, die im Atom eingetroffen sind.
- Output: die Zahl der Produkte, die das Atom verlassen haben.
- Average staytime: die durchschnittliche Verweildauer der Produkte im Atom.

Wenn Sie im Menü Results die Option Report wählen, erscheint im Fenster ein Report wie in Abbildung 5-6.

summary report

name	content		throughput		staytime average
	current	average	input	output	
Source1	1	0.941	211	210	160.976
Saw	10	8.952	2100	2090	153.742
Queue 1	95	84.847	2090	1995	1489.164
Milling machine	1	0.995	1000	999	35.837
Milling machine	1	0.995	995	994	36.019
Queue 2	1	0.624	1993	1992	11.269
Bench vice 1	4	3.763	996	992	136.107
Bench vice 2	4	3.761	996	992	136.304
Storage	0	0.000	496	496	0.000
Sink11	0	0.000	496	0	0.000
Product	0	0.000	0	0	0.000

Model start time Thursday, March 05 2009 14:17:27
Model end time Friday, March 06 2009 00:17:27
Runlength (seconds) 36000.00

Abbildung 5-6: Summary report

So lässt sich in Abbildung 5-6 ablesen, dass ein Produkt durchschnittlich 36,038 Sekunden in Verschlussbank eins blieb, dass in Verschlussbank zwei 248 Produkte produziert wurden und noch 4 vorhanden sind. Berechnen Sie ausgehend von diesem Report, dass im Schnitt 49,6 Fenster pro Stunde hergestellt wurden!

Mit der zweiten Option Grafiken ist es möglich, eine grafische Darstellung einer Variablen zu geben. Für diese Methode ist es erforderlich, die Atom History für das Atom, für das die Grafik erstellt werden soll, zu aktivieren. Dies geschieht über die Menüoption History im Menü Simulate.

Achtung! Sowohl die Option „General history“ (durch Aktivieren in der Checkbox) als auch die separaten Atome (über die Option „One on“) müssen ausgewählt werden (siehe auch Abbildung 5-7).

Die Auswahl der Option „All on“ erscheint zunächst praktisch, da dann stets von allen Atomen eine History gespeichert bleibt. Allerdings wird hierdurch, vor allem bei größeren Modellen, die Simulation unnötig langsam.

An dieser Stelle wird lediglich auf die Möglichkeit der Erstellung von Grafiken mit Hilfe der Menüoption Graphs eingegangen, da mit dem Graph Atom später programmiert werden soll. Für fortgeschrittene Anwender ist das Graph Atom sehr praktisch, da mit Hilfe von 4DScript alle möglichen Größen grafisch dargestellt werden können.

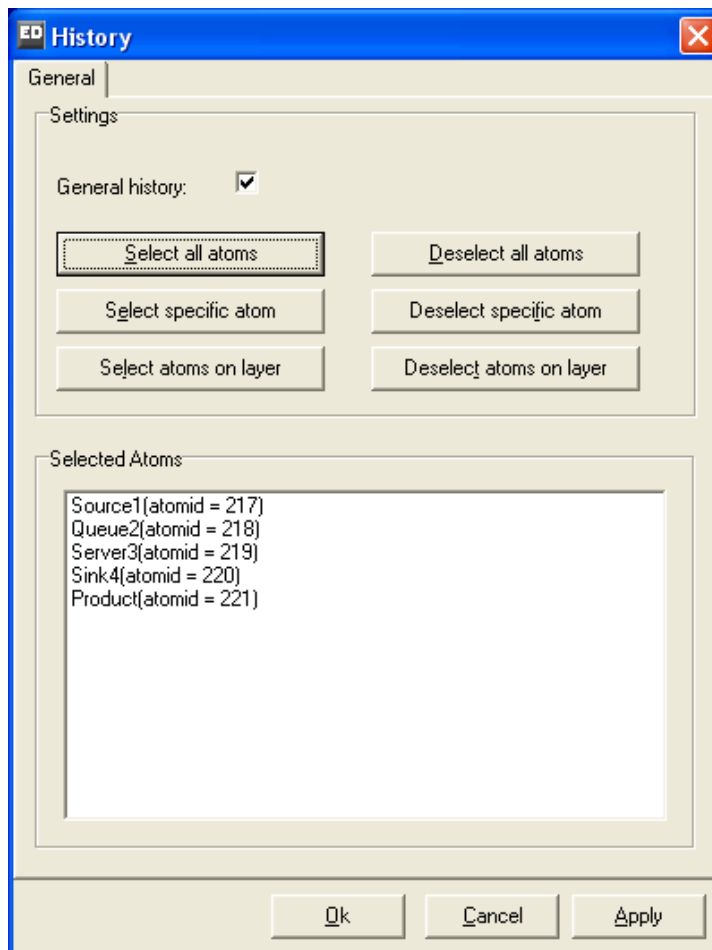


Abbildung 5-7: History

Wenn Sie die Option Graphs im Menü Results anklicken, erscheint ein Fenster, mit dem das in der Grafik darzustellende Atom ausgewählt wird. Wenn ein Atom ausgewählt ist, erscheint das in Abbildung 5-8 gezeigte Fenster.

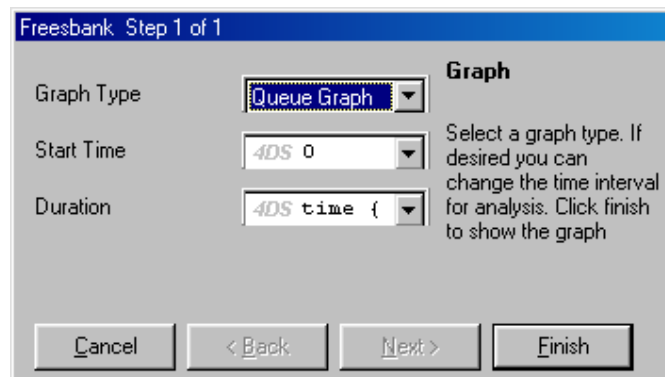


Abbildung 5-8: Eingabefenster Graph

Die beiden letzten Eingabefelder besprechen wir zuerst:

- **Start time**
Hier wird der Simulationszeitpunkt angegeben, bei dem die Grafik beginnt. Steht hier eine 0, so beginnt die Grafik mit Simulationsstart. Wenn hier der Wert 10 steht, werden die Messungen der ersten 10 Sekunden in der Grafik nicht berücksichtigt.
- **Duration**
Hier wird die Zeitspanne der Grafik definiert. Wenn hier der Wert 3600 steht, wird die erste Stunde nach der Startzeit in der Grafik dargestellt.

Beim ersten Eingabefeld Graph Type wird der Typ der Grafik ausgewählt. Sie haben die Auswahl aus folgenden 5 Möglichkeiten:

1. Queue Graph.

Hiermit wird eine Grafik der Zahl der Produkte über der Zeit in einem Atom erstellt. Bei einem Server Atom wird die Zahl der Produkte in der Regel nie höher als 1 sein, bei einem Queue Atom hingegen wird die Warteschlange in der Grafik dargestellt. Abbildung 5-9 zeigt eine Grafik des ersten Buffers aus dem Zimmerei-Fallbeispiel.

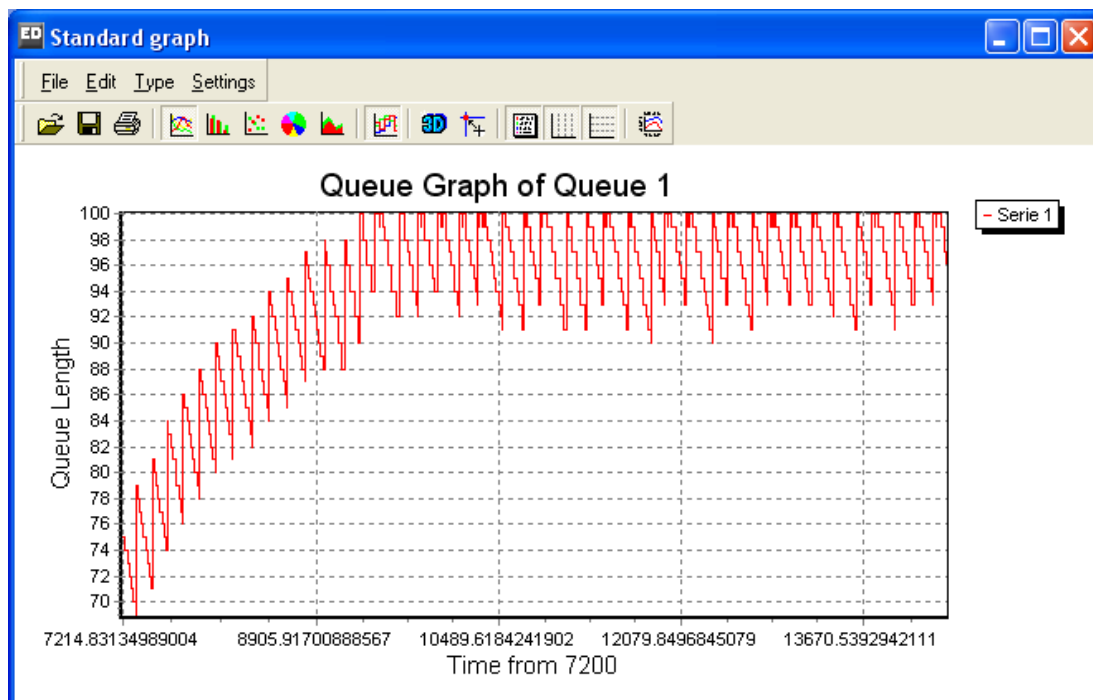


Abbildung 5-9: Queue graph

2. Hiermit Queue Histogram, hier wird jeweils der Anteil aller vorkommenden Füllstände dargestellt. Siehe Abbildung 5-10.

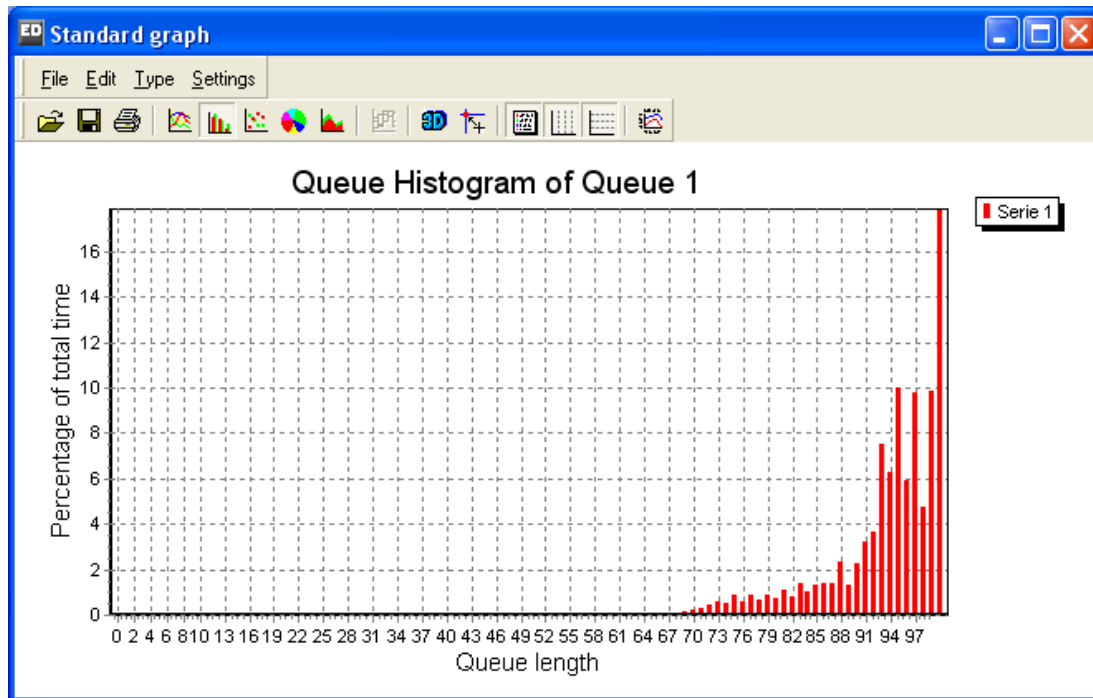


Abbildung 5-10: Queue histogram

3. Status Pie, hierin werden die Zeitanteile der Stati des Atoms in einem Kreisdiagramm dargestellt. Siehe auch Abbildung 5-11, in der ein Status Pie für die erste Fräsmaschine des Zimmerei-Fallbeispiels erstellt wurde.

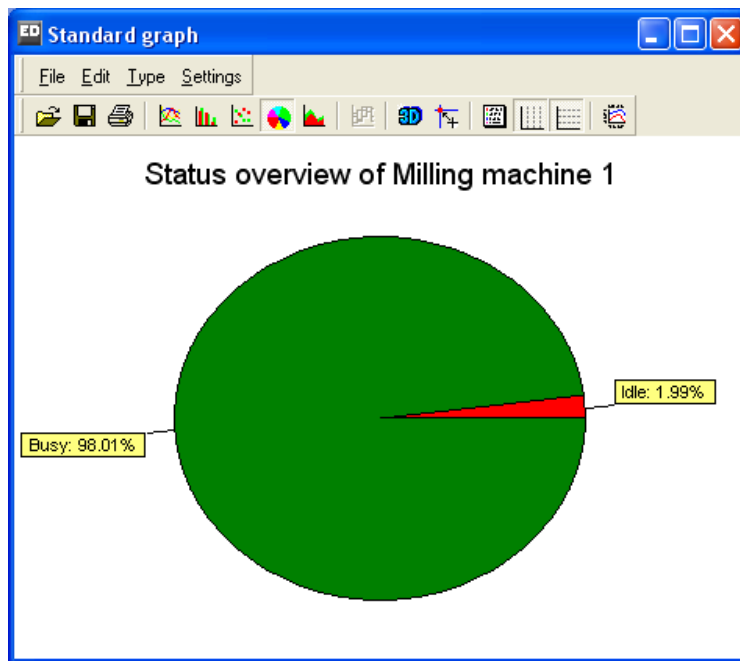


Abbildung 5-11: Status Pie

4. Hiermit Status Bar, mit dieser Option werden auch die Anteile der Stati des Atoms dargestellt, allerdings in Form eines Balkendiagramms. Siehe Abbildung 5-12.

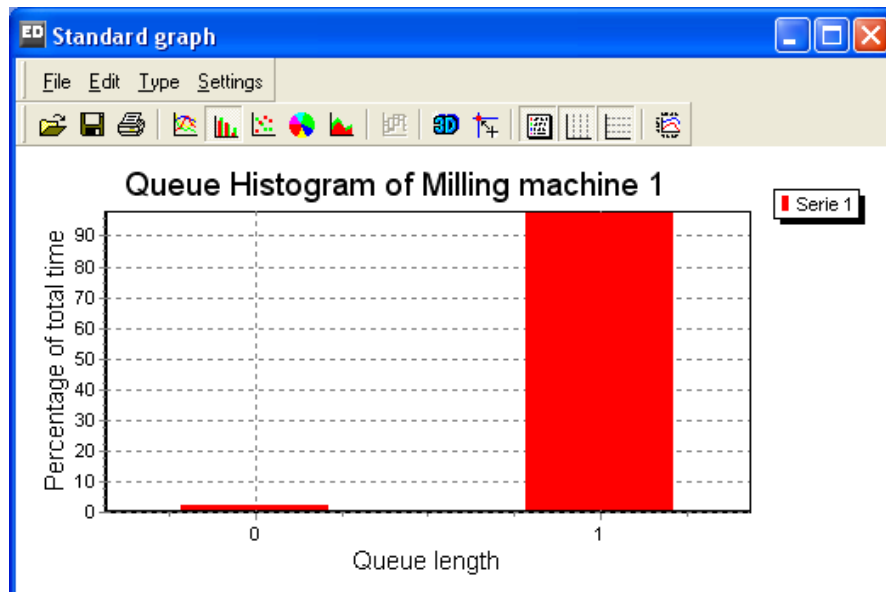


Abbildung 5-12: Status Bar

5. Hiermit Wait Histogram, hierbei wird ein Histogramm der Verweildauer der Produkte im Atom erstellt. Siehe Abbildung 5-13.

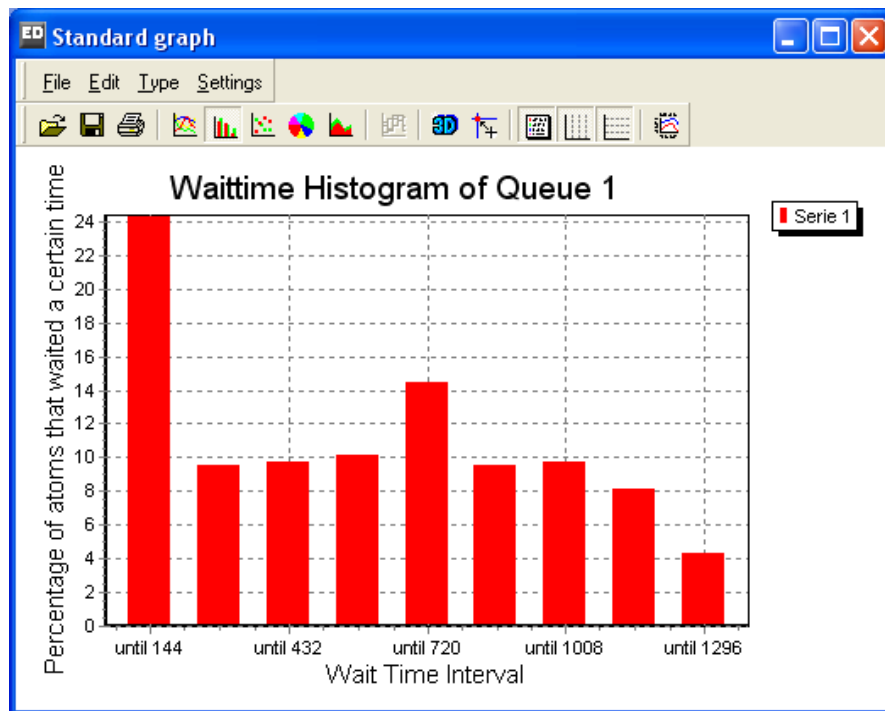


Abbildung 5-13: Wait Histogram

Zu 4. Das Experiment

Die drei Methoden der Ergebnismessung, die oben erörtert wurden, eignen sich vor allem dazu, während der Simulation sofort Ergebnisse zu sehen. Für Schlussfolgerungen bezüglich des Systems eignen sich die obigen Methoden weniger gut. Das Experiment ist hierfür die beste Methode.

Bis zur Version 5 erfolgte dies mit zwei Atomen: dem Experiment Atom und einem oder mehreren PFM Atomen (PFM = Performance Measure) als Ausgabevariablen. In Version 6 nutzen wir jedoch die Option Experimentation aus dem Hauptmenü. Diese Option ist benutzerfreundlicher, leistungsstärker und wird zudem umfassend beschrieben. Diese Beschreibung finden Sie im Dokument `experiment.pdf` unter Help | Tutorials.

Im Falle eines alten Modells mit Experiment und PFM Atomen empfehlen wir eindringlich, diese Atome zu entfernen und dasselbe Experiment mit dem neuen Experiment Wizard neu zu generieren: es dauert ein paar Minuten, lohnt sich aber unbedingt!

Wichtig ist der Unterschied zwischen den Optionen Simulate und Results auf der einen Seite und Experimentation auf der anderen: die erste Option nutzen wir im Prozess der Generierens und Testens sowie für die ersten Ergebnisse in einem *einzelnen* Durchlauf, während Experimentation im späteren Verlauf des Prozesses eingesetzt wird: wir vertrauen mehr oder weniger der Gültigkeit unseres Modells und wünschen nun Ergebnisse für *mehrere* Durchläufe, weil wir keine Schlussfolgerungen auf das Ergebnis eines einzelnen Durchlaufs basieren möchten.

Aufgabe zum ersten Casus (Fortsetzung)

5. Entwerfen Sie ein Experiment mit einer Aufwärmzeit von zwei Tagen und einem Messintervall von einer Woche, wobei mit 95%-iger Wahrscheinlichkeit die durchschnittliche Fensterproduktion pro Zeiteinheit (Stunde oder Tag oder Woche) ermittelt werden kann. Stimmt dies mit den zuvor erstellten eigenen Berechnungen und den Ergebnissen der anderen Methoden der Ergebnismessung überein?

Wir zeigen nun Schritt für Schritt die Lösung für Frage 5...

Die Experimentation umfasst vier Schritte:

- Schritt 1 Definition der Experiment Settings
Hier definieren wir Experimenteinstellungen, wie zum Beispiel die Zahl der Durchläufe, die Länge des Messintervalls und die Aufwärmzeit etc.
- Schritt 2 Definition der Performance Measures
Hier definieren wir die Ausgabevariablen eines Atoms oder einer Gruppe von Atomen
- Schritt 3 Experimentation
Die tatsächliche Simulation gemäß den Experimenteinstellungen.
Dieser Schritt erfordert keine Handlung seitens des Anwenders!
- Schritt 4 Report Definition und Ausgabeanalyse

Öffnen Sie in Experimentation den Experiment Wizard: tragen Sie anschließend die gewünschte Dauer des Messintervalls, die Zahl der Durchläufe und die Aufwärmzeit in den Experiment Settings ein (siehe Abbildung 5-14).

Experiment wizard

Experiment Settings

Settings

Simulation method: Separate runs

Observation period [s]: 4DS hr(100)

Number of observations: 5

Warm-up period [s]: 4DS hr(10)

Use terminating cond.: ☐

Condition expression: 4DS

Triggers

On start of run: 4DS

On end of run: 4DS

On end of warm-up: 4DS

Description

Experimentation for Example 2

Help Cancel < Back Next >

Abbildung 5-14: Experiment Settings

Definieren Sie auf der Sink eine Ausgabevariable (Performance Measure oder PFM) mit der Bezeichnung „Produced per hr“ (siehe Abbildung 5-15). Den Input über 100 Stunden teilen wir durch 100, um die Zahl der Fenster pro Stunde zu ermitteln.

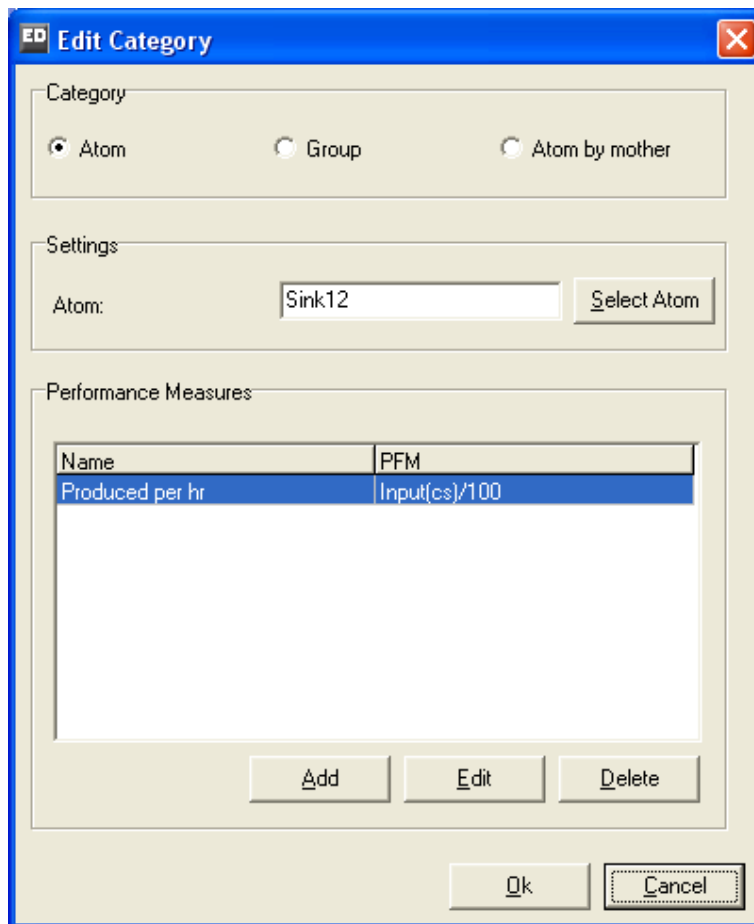


Abbildung 5-15: die gewünschte PFM mit Bezeichnung und Definition

Die tatsächliche Simulation erfolgt – nach dem Start des Experiments – in Schritt 3 (siehe Abbildung 5-16) und kann einige Zeit beanspruchen: in jedem Durchlauf von 100 Stunden müssen 500 Fenster aus 2000 Brettern und 200 Balken hergestellt werden!

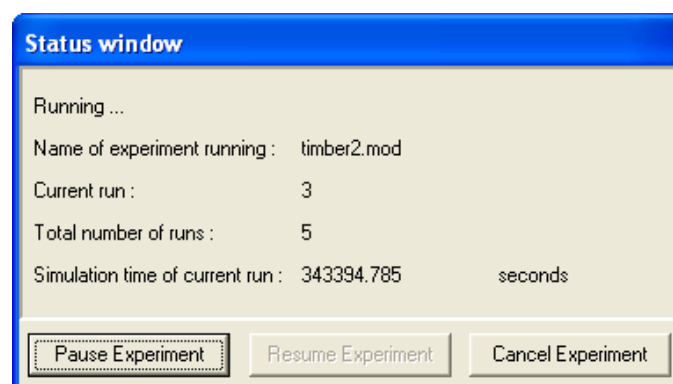
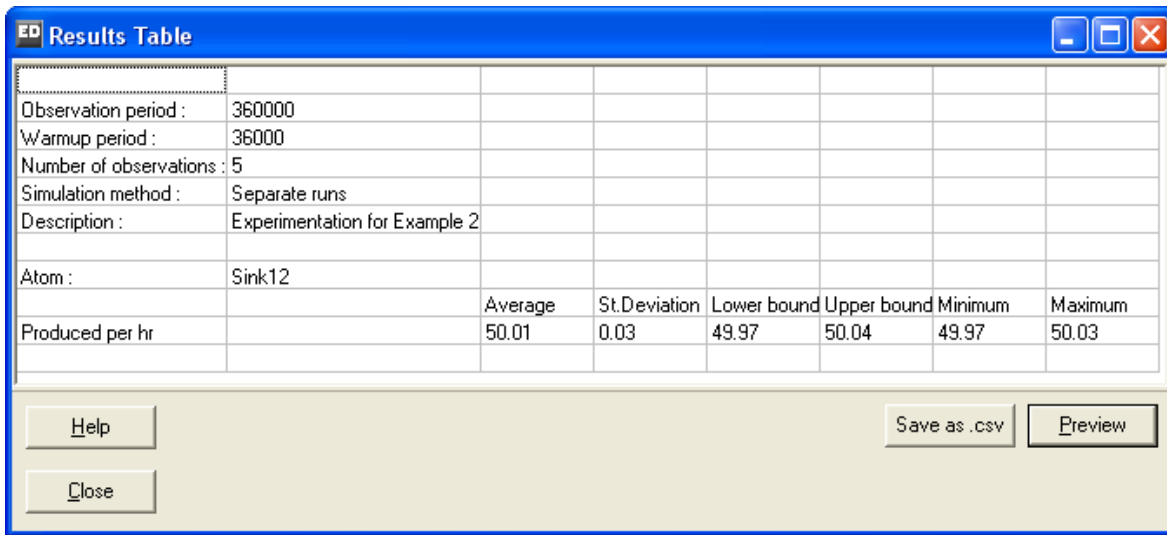


Abbildung 5-16: der Status des Experiments während der Simulation

Nach der Ausführung des Experiments kann leicht ein Report erstellt werden. Siehe Abbildung 5-17 für einen Report in einer Tabelle. Aus dieser Tabelle können Ergebnisse in Excel exportiert

(save Table as .csv anklicken und danach in Excel importieren) oder auf ansprechendere Weise über QuickReport präsentiert werden (siehe Report Preview)



The screenshot shows a window titled 'Results Table' with a blue title bar. Inside, there is a table with simulation parameters and results. The parameters section includes Observation period, Warmup period, Number of observations, Simulation method, and Description. The results section shows the Atom (Sink12) and the Produced per hr with statistical data: Average, St.Deviation, Lower bound, Upper bound, Minimum, and Maximum. At the bottom, there are buttons for Help, Close, Save as .csv, and Preview.

Observation period :	360000						
Warmup period :	36000						
Number of observations :	5						
Simulation method :	Separate runs						
Description :	Experimentation for Example 2						
Atom :	Sink12						
		Average	St.Deviation	Lower bound	Upper bound	Minimum	Maximum
Produced per hr		50.01	0.03	49.97	50.04	49.97	50.03

Abbildung 5-17: ein erstes Ergebnis in Results Table

Wir lesen ab, dass mit 95%-iger Sicherheit die unbekannte durchschnittliche Produktion pro Stunde zwischen 49,99 und 50,01 Fenstern liegen wird, eine Antwort, auf die wir inzwischen in verschiedener Form gestoßen sind.

Weitere Informationen zu den Experimenten, insbesondere zur Terminologie und zur genauen Definition unserer Standard-Ausgabeveriablen, finden Sie in unserem separaten Tutorial experiment.pdf unter Help| Tutorials.

Dieses Modell ist unter der Bezeichnung timber2.mod mitgeliefert.

6 SPIELEN MIT STRATEGIEN

In diesem Abschnitt stehen die (vordefinierten) Strategien im Mittelpunkt, mit denen Produkte Zugang zu einem nächsten Atom erhalten (Inputstrategy), aus einer Warteschlange ausgewählt werden (Queuediscipline) und nach Bedienung weitergeschickt werden (send to).

Um dies zu demonstrieren, wurde ein einfaches Modell generiert, bestehend aus drei Sources, fünf Servern, vier Queues und einem Schalter (ein selbst erstelltes Atom; Enterprise Dynamics bietet die entsprechende Flexibilität) zum Ein- oder Ausschalten der Server. Es ist keine Sink enthalten, alle Produkte bleiben also in der letzten Queue (siehe auch Abbildung 6-1).

Das Modell ist unter der Bezeichnung `strategy.mod` mitgeliefert.

Achtung: in Anlage 3 werden die wichtigsten Atome mit ihren Funktionalitäten, darunter diese Strategien, vollständig beschrieben. Nutzen Sie diese Anlage als Referenzmaterial zu diesen Illustrationen und später auch als Nachschlagewerk beim Generieren eigener Modelle!

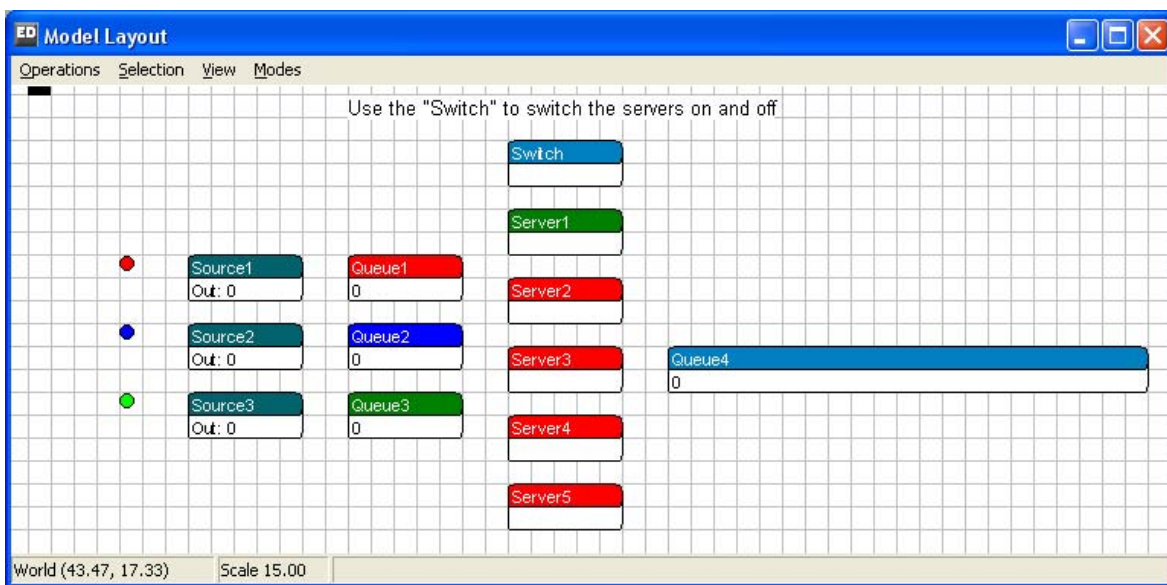


Abbildung 6-1: das Layout von `strategy.mod`

Achtung: weil im Quellcode des letzten Warteschlangen-Atoms eine Anpassung vorgenommen wurde und ein neu entworfenes Atom verwendet wird, ist es für Anwender auf Anfängerniveau nicht möglich, dieses Modell nachzubauen.

Fortgeschrittene Anwender können das Modell durchaus selbst generieren. Sie verwenden die Standardatome und passen für die letzte Queue den Code des On 2D Draw Eventhandler so an, dass in der Queue 100 Produkte sichtbar gemacht werden. Da der Schalter in diesem Fallbeispiel auch kein Standardatom ist, müssen fortgeschrittene Anwender die Server manuell ein- und ausschalten. Dies ist unter anderem mit den Availability Control Atomen möglich.

Von den drei Sources aus strömen drei Kundentypen zu den dahinter liegenden Warteschlangen. Der Zustrom ist so eingestellt, dass sich die drei linken Warteschlangen recht schnell füllen und dass nicht mehr als 10 Kunden pro Source generiert werden. Von der Warteschlange aus werden die Kunden von einem Server bedient, um anschließend in der letzten Warteschlange zu landen. Der *Schalter* dient zum Ein- und Ausschalten von Servern.

Von den drei Warteschlangen aus können im Prinzip alle Server erreicht werden, der erste Server kann jeweils über Kanal 1 erreicht werden, der zweite über Kanal 2 etc. In der ursprünglichen Einstellung von *strategy.mod* ist jedoch (über den Schalter) nur Server 1 eingeschaltet. Dies ist durch das grüne Aufleuchten des ersten Servers bei einer Simulation und das rote Aufleuchten der anderen sichtbar. Ansonsten wurden einstweilen die meisten Standardeinstellungen gehandhabt, d.h. dass

1. alle Inputstrategien noch auf *Any inputchannel* stehen
2. alle Wartedisziplinen noch auf *First in first out* stehen
3. das send to Statements auf den Queues auf 4. A random open channel: choose a random channel from all the open output channels.

Dies bedeutet, dass unter den eingeschalteten grünen Servern ein unbesetzter Server gesucht wird.

Die Produkte sind mit ihrem Icon in der letzten Queue sichtbar. Es ist also genau zu sehen, in welcher Reihenfolge die Produkte in der Queue eingetroffen sind (sofern die Queue discipline auf FIFO eingestellt ist).

Da die Produkte in dem Augenblick, wo sie die Source verlassen, ein so genanntes *Label* erhalten, können sie auch auf der Grundlage dieses Labels in der letzten Queue sortiert werden. Nebenbei, Labels haben meist die Funktion von Etiketten, die an Produkten hängen. Sie können eine Farbe, ein Gewicht, eine Größe oder dergleichen darstellen. Sie werden in 4DScript, der Programmiersprache von Enterprise Dynamics, definiert und spielen in den meisten Modellen eine bedeutende Rolle. In Abschnitt 7 wird auf den Einsatz von Labeln tiefer eingegangen.

Betrachten Sie selbst die übrigen Einstellungen von *strategy.mod* und erweitern Sie Ihr Verständnis durch Ausführung einiger Simulationsdurchläufe.

Nun kann das tatsächliche Experimentieren mit den Strategien beginnen...

6.1 Die Anpassung der Inputstrategie

Hiermit wird der Zugang zu einem Atom von vorherigen Atomen aus (mit Ausgangskanälen zu diesem spezifizierten Atom) geregelt. Durch die Inputstrategie werden ein oder mehrere Eingänge freigegeben, zudem wird über eine bestimmte Reihenfolge untersucht, über welchen Kanal zuerst Produkte aufgenommen werden können.

Die Inputstrategie lässt sich am besten mit einer Regelstrategie einer Ampel vergleichen, wobei unabhängig vom Verkehrsaufkommen für eine oder mehrere Zufahrtsstraßen aus dem Zustand Rot zu Grün umgeschaltet wird und zudem der Vorrang der Abwicklung für diese Zufahrtsstraßen festgesetzt ist.

Die ersten drei Inputstrategien öffnen alle Eingangskanäle, die beiden letzten öffnen jeweils einen Eingangskanal.

Als Übung werden nun (ausgehend vom unveränderten Anfangsmodell) die Folgen der Änderung der Inputstrategie auf Server 1 untersucht. Ändern Sie jeweils die Inputstrategie, wie nach-

folgend beschrieben, und prüfen Sie durch (wiederholte) Ausführung eines Simulationsdurchlaufs, was geschieht.

1. *Any inputchannel*

Sofern aktiviert, öffnet diese Strategie alle Eingangskanäle eines Atoms. Wenn mehr als eines der – über den Einfuhrkanal verbundenen – Atome senden will und kann, dann hat das Atom mit der niedrigsten Nummer als Eingangskanal Vorrang. So lange also immer Produkte über den ersten Kanal eintreffen, bleiben die anderen Kanäle unberücksichtigt!

2. *Largest queue*

Sofern aktiviert, öffnet diese Strategie alle Eingangskanäle eines Atoms. Wenn mehr als eines der – über den Einfuhrkanal verbundenen – Atome senden will und kann, dann hat das Atom mit der längsten Warteschlange oder dem größten Inhalt Vorrang. Achten Sie darauf, dass im Falle mehrerer gleich langer Warteschlangen jeweils der Eingangskanal mit der niedrigsten Nummer gewählt wird.

3. *Longest waiting*

Sofern aktiviert, öffnet diese Strategie alle Eingangskanäle eines Atoms. Wenn mehr als eines der – über den Einfuhrkanal verbundenen – Atome senden will und kann, dann hat das Atom mit der längsten durchschnittlichen Verweildauer Vorrang. Im Falle mehrerer Atome mit derselben durchschnittlichen Verweildauer wird jeweils der Eingangskanal mit der niedrigsten Nummer gewählt. Achten Sie darauf, dass dies nicht per definitionem bedeutet, dass die Warteschlangen mengenmäßig ungefähr gleich lang werden, wie es bei der vorigen Option der Fall war.

4. *Round robin*

Öffnet zunächst den ersten Eingangskanal und wartet auf Zusendung eines Produkts über diesen Eingangskanal. Beim nächsten Zyklus kommt der zweite Eingangskanal an die Reihe etc. Wenn der letzte Eingangskanal an der Reihe war, wird das Verfahren mit dem ersten Eingangskanal fortgesetzt.

Achtung! Beim ersten Produkt, das eintrifft, gilt diese Regel noch nicht, somit wird die Produktreihenfolge in unserem Beispiel $x_2, 3, 1, 2, 3 \dots$ lauten, wobei x jedes mögliche Produkt sein kann!

5. *Channel 1.*

Hierbei wird ein fester Eingangskanal angegeben, der genutzt werden soll. In diesem Fall dürfen also nur Produkte über Eingangskanal 1 eintreffen. Bitte beachten Sie, dass diese Regel beim ersten eintreffenden Produkt noch nicht gilt und alle Kanäle offen sind.

6.2 Das Anpassen der Queue Discipline

In diesem Teil soll die Queue Discipline der letzten Queue angepasst und beobachtet werden, was passiert.

Setzen Sie die Inputstrategie des ersten Servers auf Largest Queue, weil die Produkte hierdurch weitestgehend durcheinander den Server erreichen (und folglich auch wieder verlassen). Lassen Sie ansonsten die übrigen vier Server ausgeschaltet, weil bei fünf eingeschalteten Servern weniger deutlich zu verfolgen ist, wie die Atome sich bewegen. Nachfolgend werden die sechs möglichen Einstellungen beschrieben (siehe auch Anlage 2: das Queue Atom)

1. *First in first out*

Hierdurch werden die Atome in der Reihenfolge ihres Eintreffens in der Queue platziert.

2. *Last in first out*
Hierdurch werden die eingehenden Produkte vorn in der Warteschlange platziert. Die Produkte verlassen die Queue also in der umgekehrten Reihenfolge ihres Eintreffens.
3. *Random*
Diese Wartedisziplin platziert die Produkte nach dem Zufallsprinzip in der Warteschlange.
4. *Sort by **Label** Ascending*
Hierbei werden die Produkte mit dem niedrigsten Wert für ein bestimmtes Label vorn in der Warteschlange platziert. Testen Sie dies, indem Sie nach dem Label „Product“, das auf allen Product Atomen in diesem Fallbeispiel vorhanden ist, sortieren. *Achtung:* wenn die Produkte nicht richtig sortiert werden, hängt dies wahrscheinlich mit einem Leerzeichen vor oder hinter dem Namen des Labels zusammen.
5. *Sort by **Label** Descending*
Hierbei werden die Produkte mit dem höchsten Wert für ein bestimmtes Label vorn in der Warteschlange platziert.
6. *User defined*
Hierbei werden die Produkte an einer benutzerdefinierten Position in der Warteschlange platziert. Testen Sie dies für den Wert 5.

6.3 Das Anpassen des Send to Statements

In diesem dritten Teil des Fallbeispiels werden einige „Send to“ Statements für die drei Warteschlangen zu den Servern behandelt. Mit dem „Send to“ Statement definieren wir den Ausgangskanal, zu dem wir die Produkte senden. Vom Anwender kann eine Zahl eingegeben oder eine der 21 (!) vordefinierten Optionen gewählt werden. Eine vollständige Übersicht finden Sie in der Atombeschreibung der Source in Anlage 2.

Zunächst schalten wir die fünf Server ein, sonst gibt es keine Verzweigungsalternativen. Hierdurch erhalten sie die Farbe Grün und nehmen Produkte an. Experimentieren Sie nun bei den Warteschlangen mit den folgenden Methoden:

1. *Specific channel: always send to channel 1.*
Hierbei wird das Product Atom immer zu einem festen Ausgangskanal geschickt. Wählen Sie zum Beispiel Kanal 3.
2. *An open channel (First channel first): search, starting from the first channel, and send to the first open channel found.*
Hierbei wird das Product Atom zum ersten offenen Kanal geschickt, den Enterprise Dynamics findet. Es wird beim ersten Ausgangskanal beginnend gesucht (Kanal Nummer 1).
3. *An open channel (Last channel first): search, starting from the last channel, and send to the first open channel found.*
Es wird beim letzten Kanal beginnend gesucht, und das Produkt wird zum ersten offenen Kanal geschickt, den Enterprise Dynamics findet.
4. *A random open channel: choose a random channel from all the open output channels.*
Enterprise Dynamics wählt einen offenen Kanal nach dem Zufallsprinzip. Bei langen Simulationsdurchläufen führt dies zu gleichmäßigen Auslastungsgraden, beispielsweise bezüglich einer Gruppe von Servern, an die weitergeschickt wird.
5. *By percentage: 90% of products go to channel 1, the remaining percentage go to channel 2.*
Hierbei wird ein bestimmter Prozentsatz der Produkte zu einem Kanal und der Rest zu einem anderen Kanal geschickt. Der Anwender kann die Kanäle und den Prozentsatz definieren. Versuchen Sie einmal, 75% zu Server 1 und den Rest zu Server 2 zu schicken.

6. *By user: enter your own 4DScript expression resulting in a value between 1 and the number of channels: 1. You can press the small button for the 4DScript editor.*
Ein vom Anwender geschriebener 4DScript Code, der den Ausgangskanal definiert. Wenn Sie auf den kleinen viereckigen Button neben dem Text klicken, erscheint der 4DScript Editor. Tragen Sie zum Beispiel 2 ein.

Hiermit werden die Strategien abgeschlossen. Natürlich gibt es viel mehr Möglichkeiten für die Weiterleitung, diese müssen jedoch vom Modellierer in 4DScript, der ED zugrunde liegenden Programmiersprache, programmiert werden.

7 WEITERE ATOME: VON ASSEMBLER BIS UNPACK

In den vorigen Abschnitten wurden die Basisatome erörtert. Anschließend wurde das Generieren und Experimentieren mit einfachen Modellen geübt. In diesem Abschnitt werden acht neue Atome eingeführt. Mit diesen neuen Atomen können auch komplexere Situationen modelliert werden. Daneben widmen wir den Darstellungsmöglichkeiten in ED mehr Aufmerksamkeit.

Die Atome, die wir in diesem Abschnitt besprechen, sind:

- *Assembler*
Dieses Atom wird genutzt, wenn mehrere Atome in einem neuen Atom aufgehen. Hierbei können die alten Atome erhalten bleiben (Verpacken) oder gelöscht werden (Assemblage).
- *Unpack*
Dieses Atom kann zusammengefügte Atome wieder aufteilen. Wenn mit Hilfe eines Assemblers zum Beispiel ein Karton gefüllt wird, kann Unpack diesen wieder entpacken.
- *Container*
Der Container ist ein Atom zum Platzieren oder Verpacken von Produkten. Beispiele sind ein Karton oder eine Palette.
- *Accumulating conveyor*
Der Accumulating conveyor ist ein Transportband, das zugleich als Puffer fungiert. Wenn die Produkte das Transportband nicht mehr verlassen können, läuft das Band weiter. Hierdurch werden die Produkte auf dem Transportband ineinander geschoben. Der Accumulating conveyor wird zum Modellieren von Rollenbändern eingesetzt.
- *Non-Accumulating conveyor*
Der Non-Accumulating conveyor ähnelt in starkem Maße dem Accumulating conveyor, läuft jedoch nicht weiter, wenn die Produkte das Transportband nicht mehr verlassen können. Die Entfernung zwischen den Produkten bleibt also immer identisch. Der Non-Accumulating conveyor wird zum Modellieren von Kettenbahnen genutzt.
- *MultiService*
Das MultiService Atom fungiert als Gruppe paralleler Server: es enthält die Grundfunktionalitäten des Server Atoms und kann daneben mehrere Produkte gleichzeitig und unabhängig von einander verarbeiten.
- *Lock*
Das Lock Atom lässt nur eine zuvor eingestellte Zahl von Produkten hindurch. Alle nachfolgenden Produkte werden blockiert.
- *Unlock*
Sobald ein Produkt das Unlock Atom verlässt, kann das Lock Atom wieder ein neues Produkt durchlassen. Hierdurch ist es möglich, den Vorrat laufender Arbeiten oder die Menge der Transportmittel im Modell zu managen.

All diese Atome werden im nachfolgenden Casus behandelt. Eine vollständige Beschreibung dieser Atome ist in Anlage 2 enthalten.

Casus 3 Stapeln und Wickeln

In der Versandabteilung einer Fabrik werden Produkte auf eine Palette gestapelt. Die Produkte und die Paletten kommen über zwei separate Transportbänder bei diesem Stapler oder Palet-

tierroboter an. Dieser Roboter kann die Produkte, abhängig von ihren Abmessungen, sowohl auf- als auch nebeneinander stapeln.

Vom Roboter aus wird die gestapelte Palette mit Produkten über ein nachfolgendes Transportband zu einem so genannten Wickler befördert, wo die Produkte in Plastik gewickelt werden. Am Wickler können die Produkte mehrerer Paletten gleichzeitig und unabhängig voneinander gewickelt werden.

Das System besteht aus folgenden Komponenten:

- Alle 5 Sekunden kommt über eine Rollenbahn ein Produkt beim Stapler an.
- Über eine zweite Rollenbahn kommt im Schnitt alle 40 Sekunden gemäß negativ-exponentieller Verteilung eine Palette beim Stapler an.
- Der Staplerroboter platziert jeweils 8 Produkte auf die Palette und benötigt dafür, wenn alle Produkte vorliegen, genau 20 Sekunden.
- Die vollen Paletten gelangen über eine Kettenbahn zum Wickler.
- Der Wickler kann maximal 4 Paletten zugleich mit Plastikfolie überziehen. Pro Palette sind im Schnitt 120 Sekunden (negativ-exponentiell verteilt) erforderlich, um die Produkte der Palette in Plastik zu verpacken.
- Die Länge jedes Transportbands spielt hier nicht wirklich eine Rolle: wählen Sie diese in der Größenordnung von 10 Metern. Die Geschwindigkeit beträgt jeweils 1 m/s. Die Produkte sind jeweils 50 cm lang, breit und hoch, und die Paletten sind jeweils 1 Meter lang und breit.

Wir möchten dieses System in Enterprise Dynamics modellieren:

Für die Rollenbahn verwenden wir den Accumulating Conveyor, für die Kettenbahn einen Non-Accumulating Conveyor. Daneben für den Stapler einen Assembler, für den Wickler ein Multi-service Atom und für die Palette den Container.

Aber... berechnen Sie zuvor noch, ob dieses System den Zustrom verarbeiten kann!

Die meisten Atome können problemlos im Modell platziert werden weil sie in Standardform mit anderen Atomen verbunden sind.

Beim Assembler müssen die Paletten über den ersten Kanal (dargestellt mit einem Viereck) in das Atom gelangen und die Produkte über den zweiten Kanal. Doppelklicken Sie den Assembler für die Zahl der Produkte pro Palette. Es erscheint eine Tabelle (siehe Abbildung 7-1), wo pro Eingangskanal angegeben wird, wie viele Atome zusammengefügt werden müssen. Siehe Anlage 2 für eine komplette Beschreibung aller eingesetzten Atome, darunter des Assemblers!

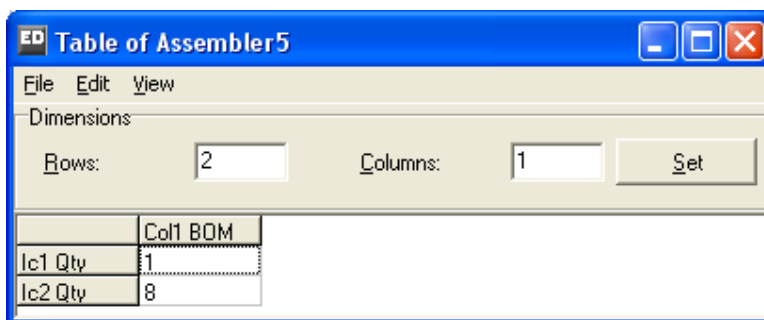


Table of Assembler5	
File Edit View	
Dimensions	
Rows:	2
Columns:	1
Set	
	Col1 BOM
lc1 Qty	1
lc2 Qty	8

Abbildung 7-1: Bill of Material des Assemblers

Die Produkte werden vom Assembler automatisch auf die Palette gestapelt. Der Platz, der für jedes Produkt reserviert wird, ist im Containeratom eingestellt. Um zu verhindern, dass alle Pro-

dukte in einer hohen Säule übereinander gestapelt werden, müssen die Abmessungen des Produkts an das Produktatom angepasst werden. Siehe hierfür die Ausarbeitung zu Casus 2!

Wenn alle Atome im Modell untergebracht sind, sieht das Layout des Modells wie in Abbildung 7-2 aus.

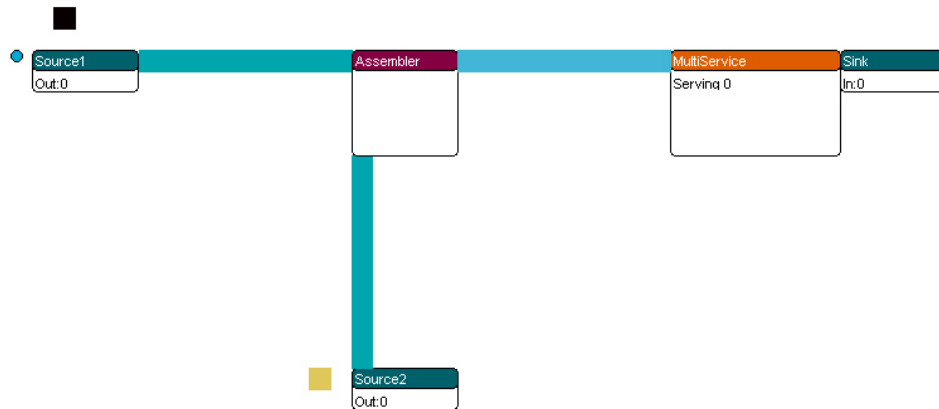


Abbildung 7-2: Layout des Conveyor-Systems

Speichern Sie das eigene Modell ab. Bei Unsicherheit bezüglich des eigenen Modells kann für die nächsten Abschnitte auch das mitgelieferte Modell mit der Bezeichnung conveyor1.mod genutzt werden.

Lassen Sie das System einige Zeit laufen und kontrollieren Sie die Funktion in 3D. Spielen Sie mit den Abmessungen der Produkte oder des Containers und schauen Sie, was passiert. Tun Sie das Gleiche mit den Geschwindigkeiten der Transportbänder oder den Bearbeitungszeiten: so ergibt zum Beispiel eine viel höhere Bearbeitungszeit beim MultiService Stauungen vor dem Wickler; dies verdeutlicht die Funktion des Non-Accumulating Conveyor.

- 1 Um die Arbeitsbelastung managen zu können, möchten wir nicht mehr als fünf Paletten zugleich in diesem System haben. Passen Sie das Modell gemäß diesem Wunsch an.

Dies wird durch die Platzierung eines Lock und Unlock Atoms im Modell erreicht. Die Lock und Unlock Atome werden im Produktionsprozess eingefügt: das Lock an der Stelle, wo die Beschränkung beginnt, das Unlock dort, wo die Beschränkung endet. Das Lock Atom schließt sich selbsttätig, nachdem es von einer zuvor eingestellten Zahl von Atomen passiert wurde. Das Unlock Atom hebt diese Blockierung wieder auf.

Wenn lediglich ein Lock platziert wird, können zum Beispiel nicht mehr als fünf Atome in das System gelangen. Indem auch ein Unlock platziert wird, sorgen wir dafür, dass nicht mehr als fünf Atome zugleich im System präsent sein können. Das Lock Atom wird stets nach der Source platziert, weil die Source sonst das Lock Atom selbst als Produkt betrachtet und in das System schickt. Das Unlock Atom wird stets an der Stelle platziert, wo die Beschränkung der Zahl der Atome wieder aufgehoben werden soll, zum Beispiel nach einer bestimmten Maschine oder kurz vor der Sink. In diesem Casus wird das Unlock Atom also unmittelbar vor der Sink platziert werden müssen.

Durch die Platzierung eines Unlock Atoms öffnet das Lock Atom sich automatisch, wenn ein Atom das Unlock Atom verlässt. Das Lock Atom und das Unlock Atom werden automatisch miteinander verbunden. Sollte es doch erforderlich sein, sie manuell zu verbinden, so kann dies durch die Verbindung des zweiten Ausgangskanals des Lock Atoms mit dem zweiten Eingangskanal des Unlock Atoms geschehen. Um die Funktion des Lock Atoms und des Unlock Atoms deutlicher zu sehen, kann das Lock Atom vorübergehend so eingestellt werden, dass es sich nach 1 oder 2 Paletten schließt. Siehe Anlage 2 für eine umfassende Beschreibung!

Passen Sie nun das Modell an oder öffnen Sie conveyor2.mod

- 2 Der Betrieb möchte nun nach dem Wickler die Paletten von den Produkten trennen und über ein zusätzliches Transportsystem zum Stapler zurück führen. Die Zahl der Paletten im System bleibt auf 5 beschränkt. Passen Sie das Modell entsprechend an.

Um die Wiederverwendung zu ermöglichen, sind zwei große Anpassungen erforderlich. Erstens die Trennung von Produkt und Palette, zweitens der Transport der Paletten zum Assembler. Für die Trennung der Palette und der Produkte wird das Unlock Atom benötigt. Nach dem MultiService platziert, leitet es die Produkte zum ersten Ausgangskanal und die Paletten zum zweiten. Indem an diesen zweiten Ausgangskanal eine Rollenbahn angeschlossen und diese mit der bereits bestehenden Rollenbahn verbunden wird, können die Paletten wieder verwendet werden.

Da die Paletten das System nun nicht mehr verlassen, entfällt das Unlock Atom. Das Lock Atom kann entfallen, wenn in der Source bei Number of Products 5 gewählt wird. Beachten Sie, dass die 40 Sekunden Zwischenankunftszeit beim Generieren der Paletten keine Rolle spielen und auch anders eingestellt werden. Wenn die zusätzlichen Rollenbahnen in das Modell aufgenommen wurden, könnte das Modell wie in Abbildung 7-3 und 7-4 (conveyor3.mod) aussehen. Die Länge der Transportbänder ist so eingestellt, dass der Rundgang der Paletten sofort visuell deutlich ist.

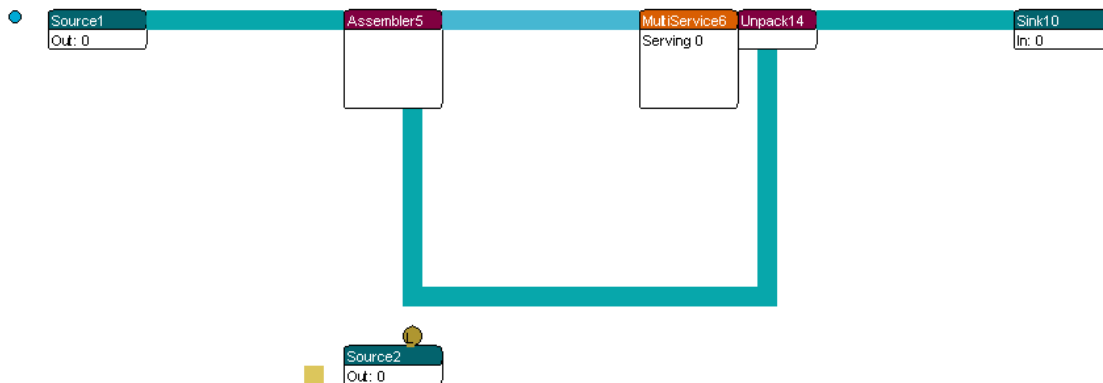


Abbildung 7-3: Wiederverwendung von Paletten in 2D

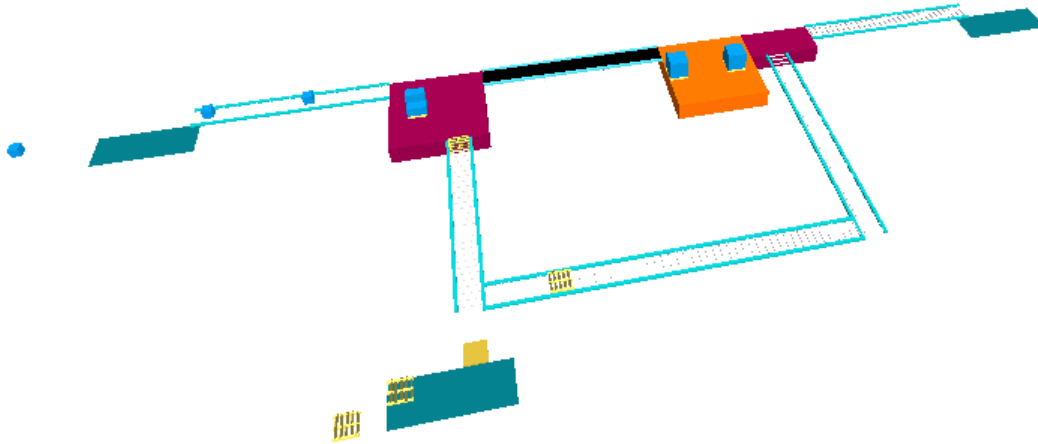


Abbildung 7-4: Wiederverwendung von Paletten in 3D

Selbstverständlich können statt der geraden Conveyors auch gekrümmte Conveyors verwendet werden. In Abbildung 7-5 wurde das Layout gespiegelt, und es wurden gekrümmte Conveyors eingesetzt, die auch höhenveränderlich sind.

Beide Modelle sind zu finden als conveyor3.mod bzw. conveyor4.mod

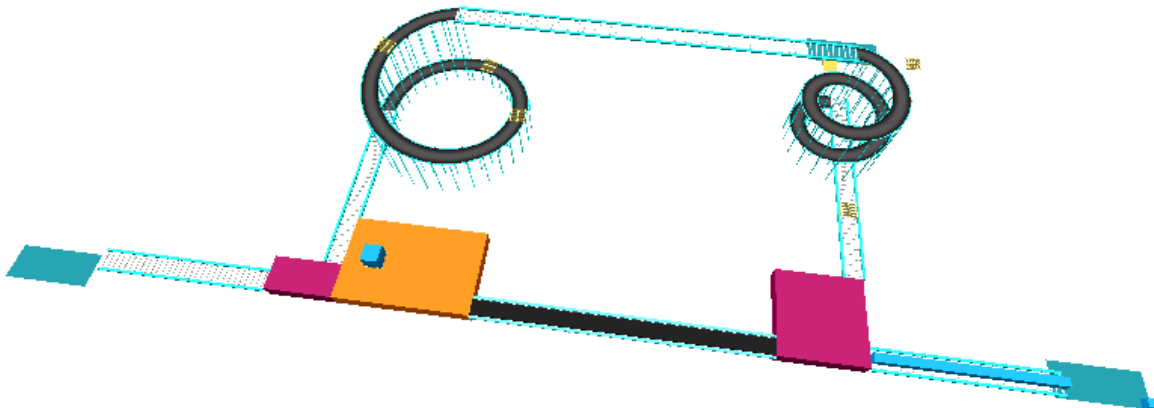


Abbildung 7-5: Höhenveränderliche Conveyors

Wir schließen mit ein paar Fragen zu conveyor3.mod:

1. Wie lange dauert eine durchschnittliche Runde einer Palette (ohne Wartezeiten)?
2. Zu wenige Paletten im System reduzieren die Produktion pro Stunde, während ab einem bestimmten Zeitpunkt zusätzliche Paletten keinen oder kaum Einfluss auf diese Stundenproduktion haben. Erklären Sie dies!
Suchen Sie die „optimale“ Zahl von Paletten, indem Sie im System experimentieren. (Betrachten Sie über einen Monitor auf Sink die durchschnittliche Stundenproduktion und variieren Sie die Menge zugelassener Paletten mit dem Lock oder über die Source.

Von Enterprise Dynamics aus können wir Verbindungen zu anderen Programmen wie Excel, Word oder Access herstellen, um Daten zu importieren oder abzuspeichern. In diesem Abschnitt wird eine häufig genutzte Anbindung zwischen ED und Excel erörtert. Dies geschieht anhand des Ausbaus eines früheren Casus bezüglich eines Postamts.

Daneben führen wir eine Reihe von Befehlen in 4DScript und das wichtige Konzept der *Label* in ED ein.

Im Anschluss an diesen Abschnitt kann der Anwender die Anbindung an Excel herstellen und beherrscht zudem die ersten Grundlagen von 4DScript.

8.1 Die Bank

Casus 4

Eine Bank in Rotterdam fungiert als Testobjekt, um neue Konzepte für die Bedienung von Kunden zu testen; hierbei gibt es mehrere Kundentypen mit verschiedenen Bedienungszeiten.

In diesem Casus beschränken wir uns – zur Illustration – auf zwei Kundentypen. So treffen nun im Schnitt 50 Kunden pro Stunde mit einer Bedienungszeit von 1 Minute ein. Diese Kunden werden als Typ A bezeichnet. Daneben treffen auch noch im Schnitt 5 Kunden des Typs B mit einer Bedienungszeit von 10 Minuten ein.

Es gibt eine gemeinsame Warteschlange, wo beide Kundentypen in der Reihenfolge ihres Eintreffens an einem von zwei Schaltern bedient werden. Alle Ankunftsprozesse nehmen wir exponentiell verteilt an. Alle Bedienungszeiten sind konstant.

Fragen und Aufgaben

- 1 Wie hoch ist der Besetzungsgrad dieses Systems?
- 2 Berechnen Sie mit Hilfe der Warteschlangentheorie die durchschnittlichen Wartezeiten und Warteschlangen, wenn beide Kundentypen eine eigene Warteschlange mit dazugehörigem Schalter haben.
- 3 Können Sie diese Merkmale auch für das System mit der gemeinsamen Warteschlange berechnen? Warum/warum nicht?
- 4 In welchem System wird die durchschnittliche Wartezeit niedriger sein? Begründen Sie Ihre Antwort!

Versuchen Sie, diese Fragen (die aus theoretischer Sicht interessant sind) zu beantworten, wir kommen später darauf zurück. Diejenigen, denen die Warteschlangentheorie für M/D/1 und M/D/2 Systeme nicht bekannt ist, können Frage 2 und 3 überspringen.

Einstweilen generieren wir zunächst das Modell mit der gemeinsamen Warteschlange. Daneben stellen wir die Verbindung zu Excel her. Siehe Abbildung 8-1 für das Layout des Modells mit den Kanälen.

Zur Unterscheidung werden die Kunden des Typs A mit einem blauen Kreis dargestellt und die Kunden des Typs B mit einem roten.

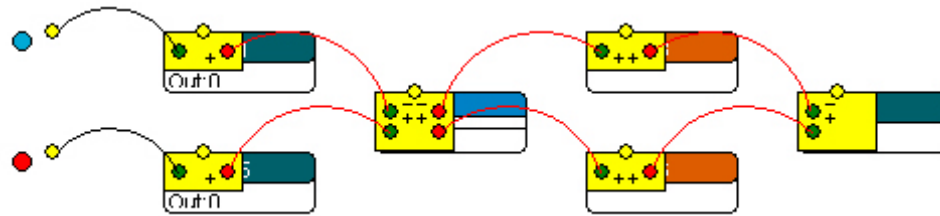


Abbildung 8-1: Layout der Bank

Auf den Servern muss nun eine Bedienungszeit ermittelt werden. Diese hängt jedoch vom Kundentyp ab, es besteht also die Auswahl aus mehreren Formeln!

Dies wird mit *Labeln* gelöst. Label sind Etiketten, die – meist – einem Produkt zugeordnet werden. Sie können eine Farbe, ein Gewicht, einen Barcode oder eine Bedienungszeit darstellen. Ein Produkt kann mehrere Label tragen, und die Zahl der Label pro Produkt kann ebenfalls variieren.

Es geht nun darum, die Bedienungszeit eines Kunden in einem früheren Stadium zu ermitteln und das Ergebnis als Label beizufügen und abzulesen, wenn die Bedienung beginnt. Das Schöne hieran ist, dass auf dem Server nicht der Kundentyp geprüft werden muss, sondern nur das Label des Kunden. Zudem lässt sich dieses Vorgehen auch leicht bei drei oder mehr Kundentypen einsetzen!

Wir hängen an jedes Produkt beim Trigger on Exit der ersten bzw. zweiten Source ein Label mit der Bezeichnung Servicetime. Dieses enthält die richtige Dauer der Bedienungszeit:

setlabel([servicetime],mins(1),i)	für Kunden vom Typ A
setlabel([servicetime],mins(10),i)	für Kunden vom Typ B

Wichtig!

Die Notwendigkeit einer weiteren Vertiefung in die zugrunde liegende Struktur von ED und der Programmiersprache 4DScript ist deutlich. In Anlage 2 steht beim Server eine Erläuterung zur Nutzung und zur Syntax der Befehle Setlabel und Label.

Aber... für weitere Einblicke in die Syntax und den Aufbau von 4DScript ist dies der geeignete Zeitpunkt, die englischsprachige Anlage 3 durchzugehen!

Tipps!

- 1 Greifen Sie weiter regelmäßig auf diese Anlage 3 zurück, um die Kenntnisse im Hinblick auf 4DScript systematisch auszubauen. Der Schritt zur Anleitung mit mehr als 1000 Befehlen wird dann ein Stück kleiner.
- 2 Ein Doppelklick in den Feldern, in denen 4DScript eingegeben werden kann, aktiviert den 4DScript Editor. Die F2-Taste ruft eine Liste mit 4DScript Befehlen und einer kurzen Erläuterung dazu auf.

Die Bezeichnung des Labels ist frei wählbar. Bis Version 5 waren die Label case-sensitive: SERVICETIME und servicetime sind dann unterschiedlich! Seit Version 6 besteht innerhalb eines Befehls jedoch kein Unterschied mehr zwischen Groß- und Kleinbuchstaben...

Bei der Cycletime des Servers wird das gewünschte Label abgefragt und als Bedienungszeit genutzt: `label([servicetime],first(c))`. Statt der direkten Eingabe einer Zahl wird also ein 4DScript Befehl verwendet, der eine Zahl ergibt.

Siehe Anlage 3 für die Syntax des Befehls Label und eine Erläuterung der Bedeutung von first(c).

Das vollständige Modell finden Sie als bank1a.mod. Dieses Modell kann leicht für ein System mit getrennten Warteschlangen modifiziert werden (bank1b.mod).

- 5 Schätzen Sie die durchschnittlichen Wartezeiten in beiden Systemen auf der Grundlage einer Reihe von Durchläufen mit 100 Stunden. Stimmen die Antworten mit der eigenen Intuition überein?

8.2 Die Anbindung an Excel

Statt ausschließlich auf die Statistiken in ED zu vertrauen, speichern wir die Rohdaten wie Wartezeiten und Durchlaufzeiten in Excel ab. Danach können wir die Analysemöglichkeiten innerhalb von Excel nutzen, um zum Beispiel Durchschnittswerte zu ermitteln.

Allgemeines Verfahren

1. Legen Sie eine neue Excel Datei im selben Verzeichnis wie das Modell an.
In den meisten Fällen wird dies *C:\Program Files\Enterprise Dynamics\Work* sein.

Vergessen Sie nicht, Excel zu beenden! In diesem Beispiel wird die Datei bank.xls genannt.

2. Platzieren Sie das Excel Atom im Modell. Dieses Atom finden Sie in der Bibliothek in der Kategorie Data oder in der Speedbar. Dieses Atom „organisiert“ anschließend die Verbindung zwischen ED und Excel. In jedem Modell ist höchstens ein Excel Atom verfügbar, und es ist nicht mit anderen Atomen über Kanäle verbunden.
3. Doppelklicken Sie das Excel Atom. Das folgende Fenster erscheint:

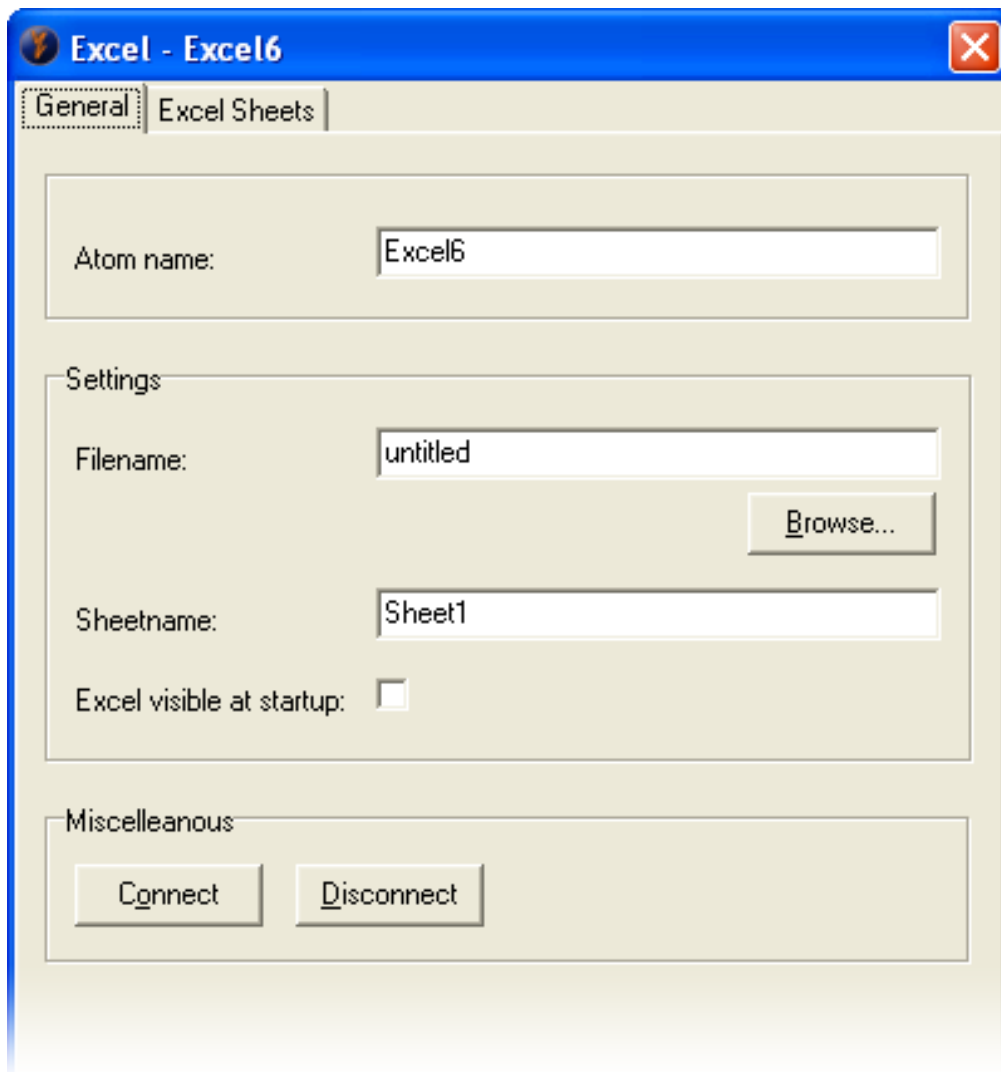


Abbildung 8-2: Das Fenster für die Anbindung zwischen ED und Excel

Wählen Sie über Browse die gerade erstellte Excel Datei aus. Wählen Sie anschließend das gewünschte Worksheet dieser Datei aus. Die Bezeichnung des hier angegebenen Worksheets muss einem der Worksheets in der Excel Datei entsprechen! Kreuzen Sie auch „Excel visible at startup“ an, um Excel im Hintergrund von ED arbeiten zu sehen.

Wenn Sie fertig sind, klicken Sie OK, und die Verbindung mit Excel wird hergestellt, wobei das Excel Icon unten am Bildschirm erscheint.

Die Anbindung ist nun hergestellt und – bei hinreichendem Auszoomen – auf dem Excel Atom sichtbar. Beim Beenden des Modells wird auch Excel beendet. Beim erneuten Öffnen dieses Modells erfolgt die Verbindung automatisch, die Schritte 1 bis 3 brauchen dann nicht jedes Mal manuell ausgeführt zu werden!

Übrigens: diese Anbindung ist eine so genannte DDE Anbindung (DDE =Dynamic Data Exchange). Hierbei müssen beide Programme, zwischen denen Kommunikation erforderlich ist, geöffnet sein. Außerdem kann nur jeweils eine Excel Datei geöffnet sein, wodurch beispielsweise keine separaten Ein- und Ausgabedateien gewählt werden können. Dies ist in ED allerdings

über ActiveX möglich, eine moderne und stabilere Form der Verbindung von zwei Programmen. Diese Option wird hier nicht erörtert.

Wenn alles nach Wunsch verlaufen ist, entsteht das folgende Layout:

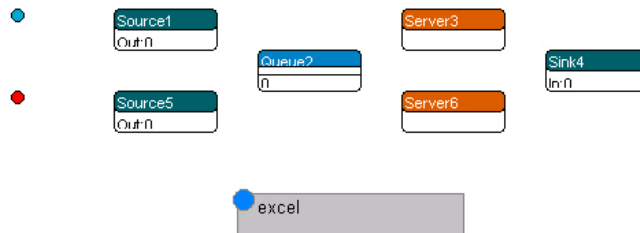


Abbildung 8-3: Die Bank mit Excel Verbindung

Bei Problemen bezüglich der Anbindung siehe Abschnitt 8.5 Troubleshooting!

8.3 Abspeichern von Daten in Excel

Um die Anbindung in diesem Casus in ihrer Funktion zu sehen, speichern wir die Wartezeit jedes Kunden in Excel ab. Dies geschieht durch den folgenden Befehl auf dem Trigger on Exit der Queue: `excelwrite(output(c),1, age(i))`

`Excelwrite(a,b,c)` speichert das Ergebnis des Ausdrucks `c` in Zelle(a,b) in der Excel Datei. Der Befehl `age` erfasst das Alter eines Produkts seit dem Eintreffen im Modell. Dieses wird hier zum Zeitpunkt des Verlassens der Queue bestimmt und entspricht deshalb der Wartezeit des Produkts!

Der Befehl `output` gibt die Zahl der Produkte an, die zu dem Zeitpunkt das betreffende Atom verlassen haben, und fungiert hier als Zähler.

Zusammengefasst: die Wartezeit von Kunde `i` wird in Zeile `i`, Spalte 1 der Excel Datei `bank.xls` abgespeichert

Stellen Sie die Geschwindigkeit der Simulation nun auf niedrig und setzen Sie die Simulation zurück. Unternehmen Sie anschließend einen Durchlauf und schalten Sie über die Taskbar von ED zur Excel Datei. Wenn alles ordnungsgemäß verlaufen ist, passiert etwas Wunderbares:

Simultan zum Prozess in ED werden die Wartezeiten in der ersten Spalte von bank.xls abgespeichert!

Verweilen Sie kurz bei diesem Augenblick: Was mit einer Information geht, geht auch mit viel mehr Informationen. Es ist also ganz leicht möglich, große Mengen Rohdaten aufzubewahren und zu speichern, um anschließend das gesamte Arsenal der Excel Funktionen darauf loszulassen!

Für diejenigen, bei denen nichts erscheint oder eine große Menge #####: schlagen Sie zunächst im Teil Troubleshooting nach. Das Problem mit vielen ##### tritt mitunter auf, wenn die Spaltenbreite unzureichend ist, um alle Daten in einer Zelle anzuzeigen. Wenn Sie die betreffende Spalte größer machen, wird dieses Problem behoben.

Für das Speichern von Durchlaufzeiten in der zweiten Spalte reicht auf dem Trigger on Entry der Sink: `excelwrite(input(c),2,age(i))`.

Achtung: Die Wartezeit von Kunde 100 steht in Zeile 100, Spalte 1, aber die Durchlaufzeit desselben Kunden 100 braucht nicht in Zeile 100, Spalte 2 zu stehen. Warum?

- 6 Ermitteln Sie in Excel die durchschnittliche Wartezeit und Durchlaufzeit der ersten 1000 Kunden

Die Modelle mit der Anbindung an Excel sind unter bank2a.mod und bank2b.mod zu finden. Die Excel Datei heißt bank.xls.

Für die Liebhaber:

- 7 Schreiben Sie die Wartezeiten von Kunden des Typs A in Spalte 3 und die Wartezeiten von Kunden des Typs B in Spalte 4 und ermitteln Sie – bei genügend Kunden – die durchschnittliche Wartezeit. Stimmen die Ergebnisse mit früheren Ergebnissen überein?

8.4 Import von Daten aus Excel

Die Anbindung von ED an Excel ist gelungen, ebenso wie das Abspeichern von Ausgabedaten, aber... der Import von Daten aus Excel kommt auch häufig vor. Dies zeigen wir im nächsten Schritt.

Im Kontext unseres Beispiels könnten tatsächlich gemessene Zwischenankunftszeiten von Kunden verfügbar sein. Dieses Ankunftsdatum möchten wir importieren und für einen Simulationsdurchlauf in ED nutzen.

Generieren Sie über eine Formel 100 (positive) Zahlen in der ersten und zweiten Spalte von bank.xls oder verwenden Sie die gerade entstandenen Wartezeiten...

In beiden Fällen empfiehlt es sich, die vorherigen excelwrite Befehle zu entfernen.

Schreiben Sie nun auf der Source für Kundentyp A bei der Interarrival-time:

`excelread(output(c)+1,1)`

Schreiben Sie für die Queue auf dem dazugehörigen Trigger on Exit:

`if(output(c)=100, closeinput(c))`

Der Befehl excelread(a,b) liest den Wert in Zelle(a,b) von Excel, während der Befehl closeoutput dafür sorgt, dass nach 100 Kunden der Ausgang des Source Atoms geschlossen wird, wodurch auch der Import gestoppt wird. Wenn dies nicht geschieht, liest ED immer weiter; in dem Fall werden leere Zellen als Zwischenankunftszeit 0 interpretiert.

Finden Sie nun selbst heraus, wie die Statements für Kunden des Typs B lauten müssen!

Dieses Modell finden Sie unter der Bezeichnung bank3.mod.

Abschließend: Ankunftsdatum werden mit einer Source generiert. Für feststehende Produktionspläne oder Dienstpläne nutzen wir in ED das Atom Arrivallist. Dieses Atom kann auch eingesetzt werden, um (große) Pläne mit dem richtigen Formal aus Excel auszuschneiden und einzufügen. Ab dem Zeitpunkt ist die Information in einer internen Tabelle verfügbar und, zumindest für die Dateneingabe, eine offene Anbindung an Excel überflüssig.

Der Import über eine Tabelle in ED sorgt für einen größeren Geschwindigkeitsgewinn und ist für die meisten Situationen denn auch sehr zu empfehlen!

8.5 Troubleshooting

Eine DDE Anbindung kann verschiedene Probleme verursachen: manchmal durch falsche Einstellungen in ED, aber auch durch die Windows Einstellungen für Excel. Die Erfahrung zeigt, dass diesem Aspekt beim ersten Mal Aufmerksamkeit gewidmet werden muss, danach wird die Verknüpfung problemlos funktionieren.

1. Prüfen Sie, welche Sprachversion von Excel Sie haben. Jede Sprache verweist nämlich in anderer Form auf Zeilen und Spalten, und dies muss in ED eingestellt werden!

Hierfür stellen Sie die entsprechenden Anfangsbuchstaben in der jeweiligen Sprache für die Wörter Spalte und Zeile unter File | Preferences ein.

In englischer Sprache (Default): C und R (**C**olumn und **R**ow), in deutscher Sprache: S und Z (**S**palte und **Z**eile), in niederländischer Sprache K und R (**K**olom und **R**ij), etc... (siehe Abbildung 8.4)

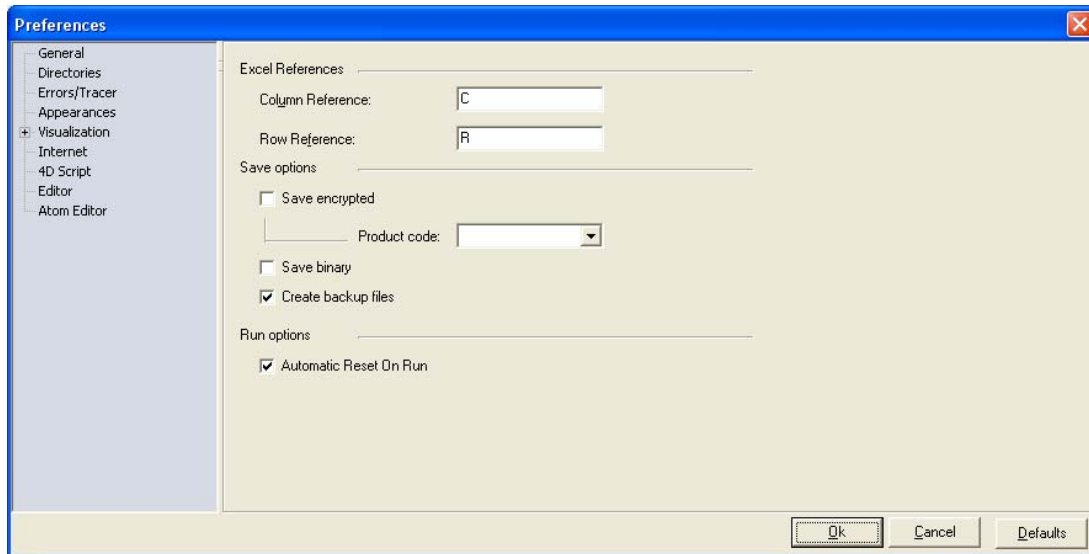


Abbildung 8-4 Das Fenster File|Preferences

2. Prüfen Sie die 3 Hinweise für die Herstellung einer Verbindung in Abschnitt 8.2:
 - a. Gibt es eine Excel Datei mit der entsprechenden Bezeichnung im Arbeitsverzeichnis? (Siehe auch Abbildung 8.3 für das Bestimmen des Arbeitsverzeichnisses)
 - b. Wurde das richtige Worksheet ausgewählt?
 - c. Wurden versehentlich mehrere Excel Dateien mit demselben Namen angelegt?

Im Zweifelsfall schließen Sie alle Excel Dateien und führen die Schritte noch einmal korrekt aus.

3. Die Verbindung wurde hergestellt, aber in Excel erscheint Zeichensalat: #####. Dies ist zurückzuführen auf die unterschiedlichen Formen, in denen Dezimalzeichen in den verschiedenen Sprachversionen interpretiert werden (Punkt bzw. Komma): Gehen Sie über Start auf Settings, Control Panel und Regional Settings|Numbers. Wählen

Sie dort für das Decimal Symbol: . (Punkt) und für das Digit Grouping Symbol: , (Komma). Wenn diese Einstellung korrekt vorgenommen wurde, kann es auch erforderlich sein, die Spaltenbreite in Excel anzupassen.

8.6 To pool or not to pool?

Die Ergebnisse des neuen Postamts zeigen etwas Bemerkenswertes: das Modell mit einer *gemeinsamen* Warteschlange (bank1.mod) weist eine höhere durchschnittliche Wartezeit auf als das Modell mit *getrennten* Warteschlangen (bank2.mod)!

Wir fand+en die folgenden Ergebnisse:

durchschnittliche Wartezeit:	gemeinsame Warteschlange (in min)	getrennte Warteschlangen (in min)
Kunden des Typs A	6.18	2.45
Kunden des Typs B	6.18	25.7
Durchschnitt für alle Kunden	6.18	4.56

Abbildung 8-5 Ergebnisse mehrerer Simulationsdurchläufe

Wie ist das nun möglich?

Im Falle getrennter Warteschlangen kann schließlich eine Schlange von Kunden des Typs A vor dem ersten Schalter entstehen, während vor dem Schalter, der für Kunden des Typs B vorgesehen ist, nicht los ist. Die Erklärung lautet: *Variation!*

Eine Kombination führt zu einer Reduzierung der durchschnittlichen Wartezeiten, wenn es um nur einen Kundentyp geht und nicht, wie in diesem Fall, zwei Typen.

Ohne allzu tief auf die Warteschlangentheorie einzugehen: bei dem System mit getrennten Warteschlangen handelt es sich um zwei unabhängige M/D/1 Modelle. Hierfür bestehen geschlossene Formeln für die wichtigsten Warteschlangenmerkmale.

Diese ergeben für die durchschnittliche Wartezeit W_q von Kunden des Typs A bzw. B:

$W_q = 2.5$ Minuten bzw. $W_q = 25$ Minuten.

Die durchschnittliche erwartete Wartezeit für alle Kunden beträgt hierdurch $50/55 \times 2.5 + 5/55 \times 25 = 4.54$ Minuten.

Bei dem System mit einer gemeinsamen Warteschlange entsteht ein M/G/k System mit zwei Kundentypen. Für die Anwendbarkeit der Formeln aus der Warteschlangentheorie „mischen“ wir die Kunden zu einem einzigen Kundentyp, von dem im Schnitt 55 pro Stunde eintreffen mit einer erwarteten Bedienungszeit von $E[S] = 50/55 \times 1 + 5/55 \times 10 = 20/11$ oder 1.81 Minuten.

Die Variation $\text{Var}[S]$ dieser Bedienungszeit beträgt 6.69 und der Variationskoeffizient C , definiert als $\text{Var}[S]/(E[S])^2$, beläuft sich auf 2.025.

Dies ergibt annäherungsweise – Formeln sind nicht gegeben – $W_q = 6.25$ Minuten.

Die Ergebnisse unseres Durchlaufs entsprechen ziemlich genau diesen theoretischen Werten: Die durchschnittliche Wartezeit bei getrennten Reihen geht um 27% zurück!

Die Erklärung? Die irregulären Kunden, deren Bedienungszeit 10-mal so lang ist wie die der regulären Kunden, beeinträchtigen bei getrennten Warteschlangen die Wartezeit der regulären Kunden nicht mehr. Obwohl dies zur Folge hat, dass die erwartete Wartezeit der irregulären Kunden sehr lang wird, reduziert sich die durchschnittliche Wartezeit für alle Kunden erheblich.

8 Aufgabe

Passen Sie das Modell der getrennten Warteschlangen so an, dass ein einseitiger Übergang für Kunden des Typs A möglich ist. Das heißt, dass der Erste in der Warteschlange des Typs A den Schalter für Kunden des Typs B nutzen kann, *sofern* keine Wartenden des Typs B da sind. Sie können natürlich auch mit einem doppelseitigen Übergang experimentieren!

Was bedeutet dies für die durchschnittliche Wartezeit für die Kunden?

Hinweis: Mit dem Zufügen von Kanälen und dem richtigen „send to“ Statement lässt sich dies relativ einfach realisieren!