Arena®

ARENA COURSE MANUAL

LISTEN. THINK. SOLVE.®

Link2

Allen-Bradley • Rockwell Software

Rockwell Automation

Copyright © 2010 Rockwell Automation, Inc. All rights reserved. This manual and any accompanying Rockwell Software products are copyrighted by Rockwell Automation, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation, Inc. is strictly prohibited.

Arena is a registered trademark of Rockwell Automation, Inc.

Microsoft and Visio are registered trademarks of Microsoft Corporation.

All other trademarks and registered trademarks are acknowledged as being the property of their respective owners.

Link-2 Ltd
No.2 Wortley Road
Deepcar
Sheffield S36 2UZ
Tel: 01142 180 472
Mob: 07476 361 106
Email: paul.binding@link-2.biz
Web: www.link-2.biz

Ver. 13.0

ROCKWELL AUTOMATION
2100 Corporate Drive, Suite 550
Wexford, PA 15090 USA
Tel 724.741.4000 Fax 724.741.4001
Support 440.646.3434
E-Mail: arena-info@ra.rockwell.com
Internet: www.arenasimulation.com

Table of Contents

1	Introduction	1
2	Introduction to Simulation	3
3	Using Basic Process Modules	9
4	Resource Sets and Schedules	37
5	Making Decisions with Simulation	73
6	OptQuest for Arena	81
7	Process Modeling	91
8	Advanced Modeling Techniques	151
A	User Interface	189
B	Debugging Tools	211
C	Commonly Used Variables and Mathematical Expressions	221
D	Expression Builder	225
E	Costing	227
F	Additional Features and Methodologies	241
G	(intentionally blank)	265
H	Statistical Distributions	279
I	Input Data & Input Analyzer	267
J	Arena Visual Designer - Dashboard	301
K	Supplemental Materials	309

1: Introduction

Welcome

Who I am and why I'm here!

Who are you and why are you here?

Course Objectives

Learn how to capture a process with simulation.

Understand what it takes to move from a description of your process to a simulation of your process.

Understand why you need to simulate.

Understand the kinds of questions and answers you can get from the use of Arena.

Leave with sound practical advice for how to succeed with simulation.

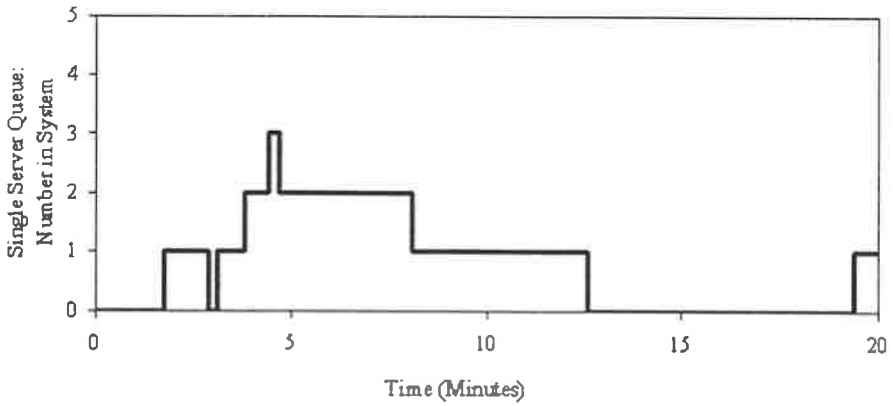
2: Introduction to Simulation

Discrete Event Simulation

A model of a real or proposed system for the purpose of evaluating the system's behavior under various conditions

In discrete event simulation we need only consider the logic of what occurs at event times, where an event is that point in time when the system status may change.

For example, consider a single server, single queue system where there are two events: an arrival and a departure.



The most powerful analysis technique for real world systems in which the dynamic and stochastic nature can be captured

Permits managers to visualize the operation of a new or existing system under a variety of conditions

Understand how various components interact with each other and how they affect overall system performance

Why Simulate?

Provide general insight into the nature of a process

Provide an understanding of how performance measures vary over time

Identify specific problems or problem areas within a system

Develop specific policies or plans for a process

Test new concepts and/or systems prior to implementation

Improve the effectiveness of a system

Provide an "insurance policy" for system performance

Simulation Considerations

Cannot give accurate results if the data is inaccurate

Cannot describe system characteristics that have not been explicitly modeled

Cannot solve problems — it can only provide information

Cannot provide easy answers to complex problems

Simulation Project Management: The Players

Simulation project team

System design team

Data/information sources

Implementation team

Contractors

Decision makers/management

Starting the Study

Have clearly stated and accepted objectives

Get input from everyone

Make certain all agendas are understood

The Simulation Process

Define — functional specification

Formulate — the simulation model

Verify/validate — input from all players

Analyze — statistical evaluation

Recommend — alternatives to the decision maker(s)

Why Have a Functional Specification?

Defines the problem completely

Requires system understanding from the start

Provides vision of the task

Defines how simulation will be used

Defines all assumptions of the simulation model

Identifies data requirements

Identifies required output statistics and analysis

Model Formulation

One or more analysts

Data structure requirements

Model control logic

Level of detail

Flexibility

Statistical requirements

Verification and Validation

Verification: Ensuring that the model behaves in the way it was intended

Validation: Ensuring that the model behaves the same as the real system

Requires

Involvement of all the players

Use of animation and data

Reasonable and robust model

The Analysis

Addresses the project objectives

An iterative process

Helps the analyst understand the results

Establishes result accuracy and sensitivity

The Final Presentation

Keep it short and simple

Answer the right questions

Address the audience

Provide reasons for the results

3: Using Basic Process Modules

Topics covered in this section:

Concepts

- Entity flow and attributes
- Capacity constraining (Resources)
- Queues

Arena Constructs

- Create logic module
- Process logic module
- Dispose logic module
- Entity spreadsheet module
- Resource spreadsheet module
- Run > Setup

Before you can build your first simulation model with Arena, there are a few fundamental concepts with which you will need to become familiar.

Terminology

Entities

The items (documents, customers, parts, etc.) that are being produced, served, or otherwise acted on by our process.

Arena is an entity-driven language. The entities represent any person, object, or thing, whether real or imaginary, whose movement through the system causes a change in the status of the system.

Examples: customers moving through a restaurant, or parts moving through a factory.

Logical entities: failure control, system control, staff breaks, etc.

Attributes

Within a system, there may be many types of entities, each having unique characteristics called Attributes.

Attributes represent user-defined values associated with individual entities, such as customer type, product size, time job entered the system, etc.

All entities have the same set of attributes (e.g., priority, arrival time) with different values.

An assignment made to an entity's attribute affects only that entity.

Some of the attributes that Arena automatically assigns are listed below. You can access and reassign these attributes when building your model:

Entity.Type — An integer value assigned to entities when they are created. Used to distinguish one entity type from another. It is automatically assigned inside each Create module.

Entity.Picture — The picture that will be displayed in the simulation animation. It is automatically assigned in the Create module (default picture is Picture.Report).

Entity.CreateTime — The simulated time that the entity was created. Used to collect cycle time information.

Entity.Station — The current station or station to which an entity is being moved. This attribute is updated automatically by Arena during a simulation run.

Entity.Sequence — Number of the entity's visitation sequence. Used when different entities need to visit the same set of stations but in a unique order.

Entity.JobStep — The entity's position or index into the sequence.

Resources

Object used to model an area where there is constraint. Constraints may be caused by limited numbers of people to perform an action, restricted storage space, capacity of equipment, etc.

The *resource capacity* is the number of identical resource units corresponding to a specific resource.

Entities *seize* resources to get control of one or more units of the resource.

Entities *release* resources when they are no longer required.

Whenever an entity seizes a resource, it must also release the same resource at some other time in the model. If the entity that controls a resource does not release it, all other entities requiring the same resource will be placed in a queue and be unable to seize the resource that will never become available.

Queues

Area where an entity resides while waiting for a resource to become available or while waiting to be batched.

Seize, Delay, and Release

In virtually every simulation model, contention for some resource must be modeled. This resource could be anything from a machine or an operator to a storage space or an intersection. These types of constraints would be modeled as resources. When using resources in Arena, you should be aware of the three main commands that are associated with the use of every resource.

Seize: Use this command whenever you want an entity to gain control of a resource. If the resource is free when an entity arrives at a Seize command, the entity will reduce the capacity of the resource by the number of resource units required in the Seize module. If the required resources are not available, the entity will wait in a queue until the resource becomes available.

Delay: Once a resource is seized, there is usually some type of action or process that must occur before an entity can release the resource. This is usually some type of delay, either explicitly defined or implied due to some other actions.

Explicitly defined delays would consist of a Delay module that defines the delay for some type of action or process. This delay can be a constant value, a random quantity from a defined distribution, or a mathematical expression. A delay can even be held by an entity attribute value that has been assigned prior to the Delay module.

Implicit delays are ones caused by other actions or conditions in the simulation model. These delays include waiting to seize another resource, waiting to be moved (or being moved) by a transporter or conveyor, or waiting for a predefined condition to occur elsewhere in the model.

Release: After the delay, the entity needs to release the resource so that it will be available for other entities within the simulation model. This action is achieved by using a Release module. Once an entity releases a resource, Arena will allow the next entity waiting for this particular resource to seize it.

Auto Loan Part 1 (Auto Loan Part 1.doe)

The following section demonstrates the basic tools used to create simulations with the Basic Process panel. The example simulates a car loan application process.

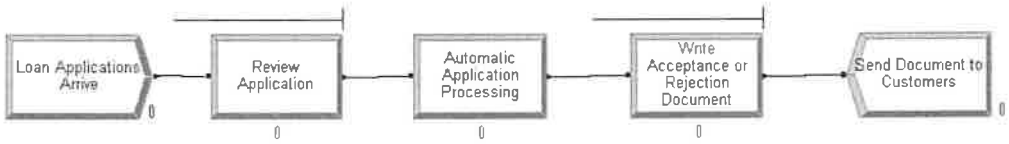
In this model, car loan applications arrive at a processing center approximately every 5 minutes. At this point, 1 of 5 loan officers will review the application to ensure that it is complete. This review step usually takes 15 minutes, but can be as short as 12 minutes or as long as 18 minutes.

Applications are sent to an automatic processing machine where they are processed. This process can take from 0.5 to 1.5 hours to complete, but usually requires 1 hour. It is assumed that the automatic processor can process as many applications as needed.

After processing, the results are reviewed by a loan officer who will write either an acceptance or rejection document. This task usually takes 7 minutes, but can require anywhere from 5 to 10 minutes to complete.

At this point, the loan application process is complete, and the acceptance or rejection document is sent to the customer.

Run this simulation for one 8-hour day.



Flowchart Modules

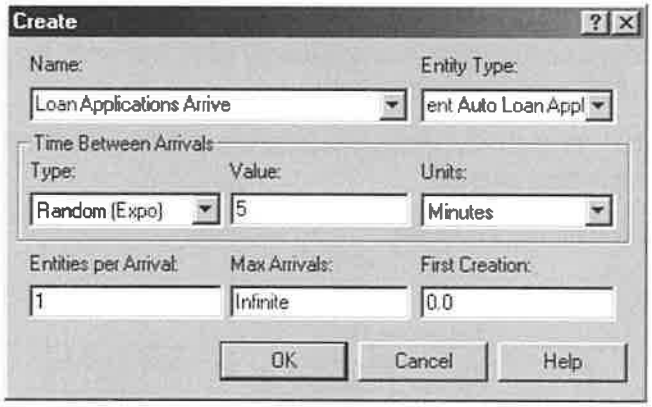
Create Module



Create

The starting point for entities in a simulation model. Entities are created using a schedule or based on a time between arrivals. Entities then leave the module to begin processing through the system.

Loan Applications Arrive – Use the Create module



Name	Loan Applications Arrive
Entity Type	Auto Loan Applications
Time Between Arrivals Type Value Units	 Random 5 Minutes
Entities per Arrival	1
Max Arrivals	Infinite
First Creation	0.0

Process Module



Process

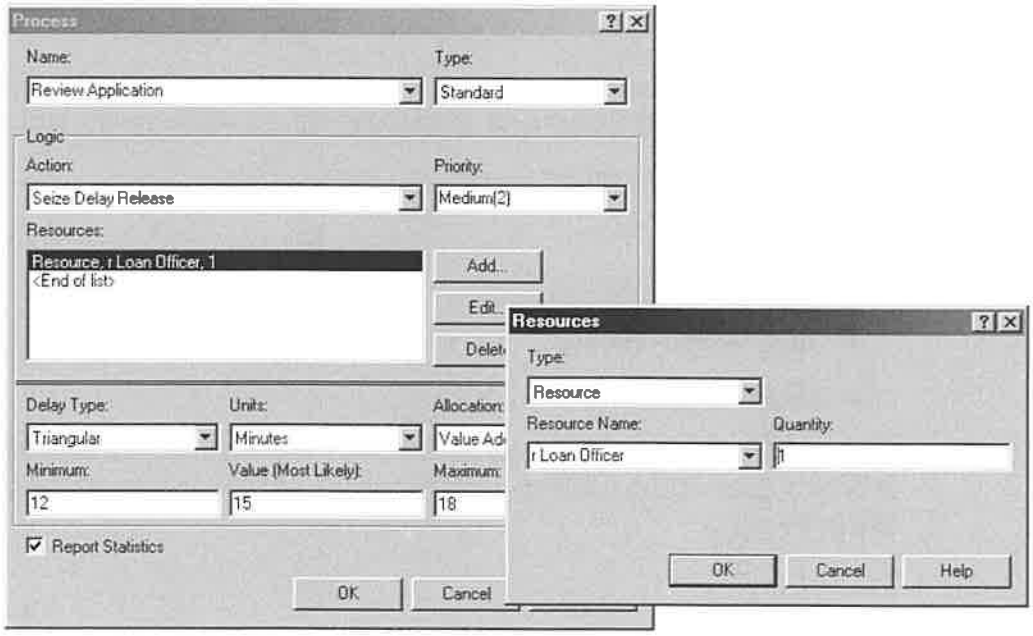
The main processing step in the simulation. Processes can be specified as value added or non-value added, so that costs may be accumulated with the entities as they process.

A Process module can be defined as one of four action options:

- Delay
- Seize – Delay
- Seize – Delay – Release
- Delay – Release

When using the Process module, it is recommended that you use it for the basic functions of Delay or Seize – Delay – Release. Other actions will be discussed in greater detail later in the course.

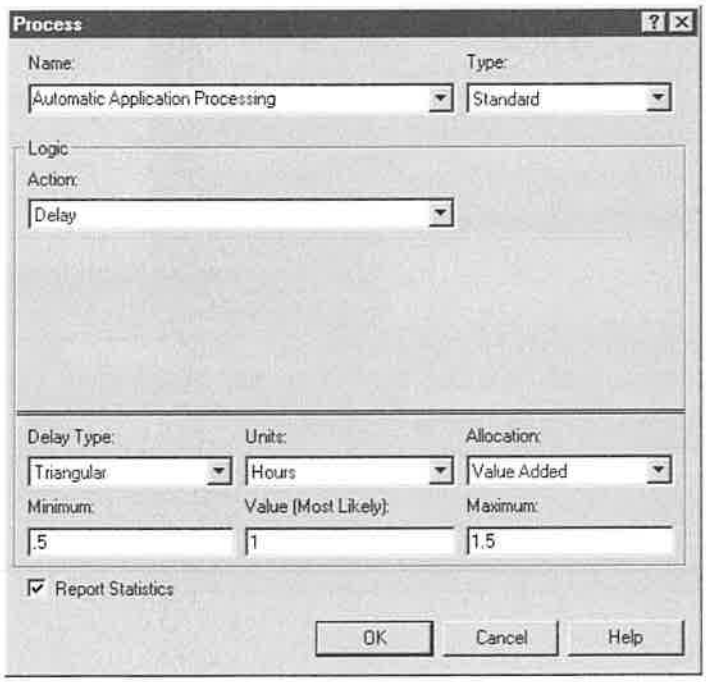
Review Application – Use the Process module



Name	Review Application
Type	Standard
Action	Seize Delay Release
Priority	Medium(2)
Resources Quantity	r Loan Officer 1
Delay Type	Triangular (12,15,18) minutes

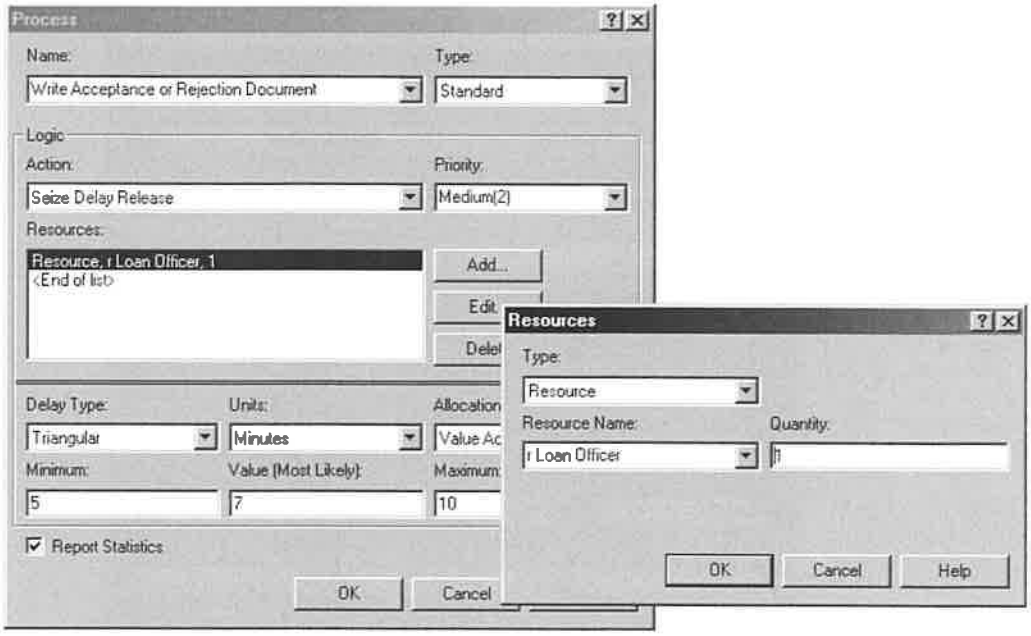
NOTE: The Resources Quantity is the number of resources to be seized for the particular process operation.

Automatic Processing Application — Use the Process module



Name	Automatic Application Processing
Type	Standard
Action	Delay
Delay Type	Triangular (.5,1,1.5) hours

Write Document — Use the Process module



Name	Write Acceptance or Rejection Document
Type	Standard
Action	Seize Delay Release
Priority	Medium(2)
Resources Quantity	r Loan Officer 1
Delay Type	Triangular (5,7,10) minutes

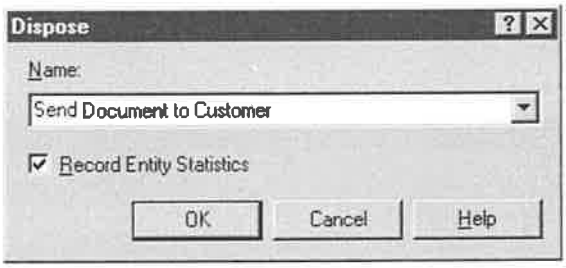
Dispose Module



Dispose

Used as the ending point for entities in a simulation model. Entity statistics may be recorded before the entity is disposed.

Completed Applications — Use the Dispose module



Name	Send Document to Customer
Record Entity Statistics	*check*

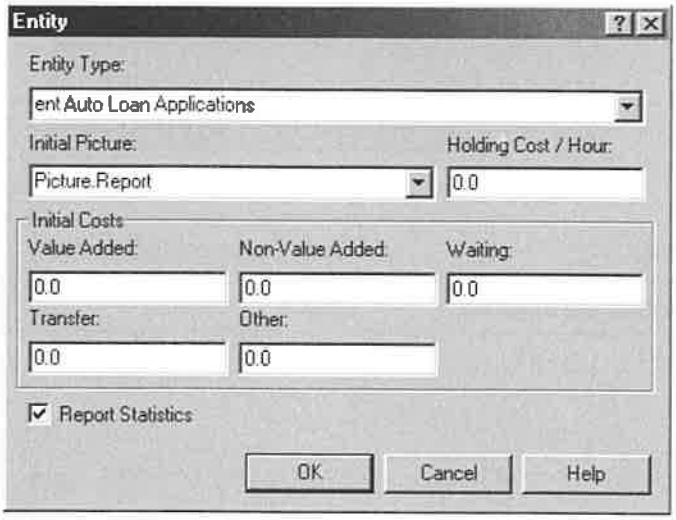
Data Modules

Entity Module



Defines the various entity types and their initial picture values in a simulation. Initial costing information and holding costs may also be defined.

NOTE: Data modules may be edited using either a dialog or the spreadsheet interface. To access the dialog, right-click with the mouse when hovering over the desired item on the spreadsheet. A menu will appear giving you several options. Select "Edit via Dialog."



Entity - Basic Process

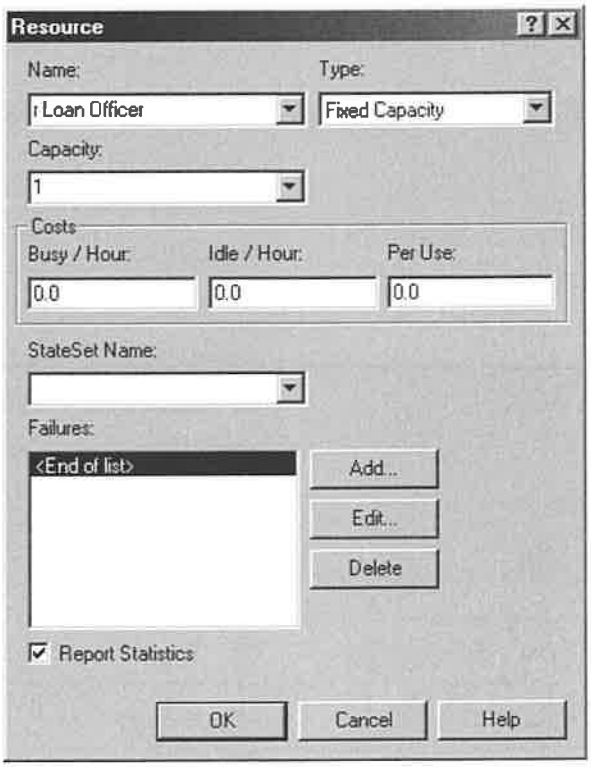
	Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran Cost	Initial Other Cost	Report Statistics
1	ent Auto Loan Applications	Picture.Report	0.0	0.0	0.0	0.0	0.0	0.0	☑

Resource Module



Resource

Defines the resources in the simulation system, including costing information and resource availability. Resources may have a fixed capacity that does not vary over the simulation run or may operate on a time-based schedule.



Resource - Basic Process

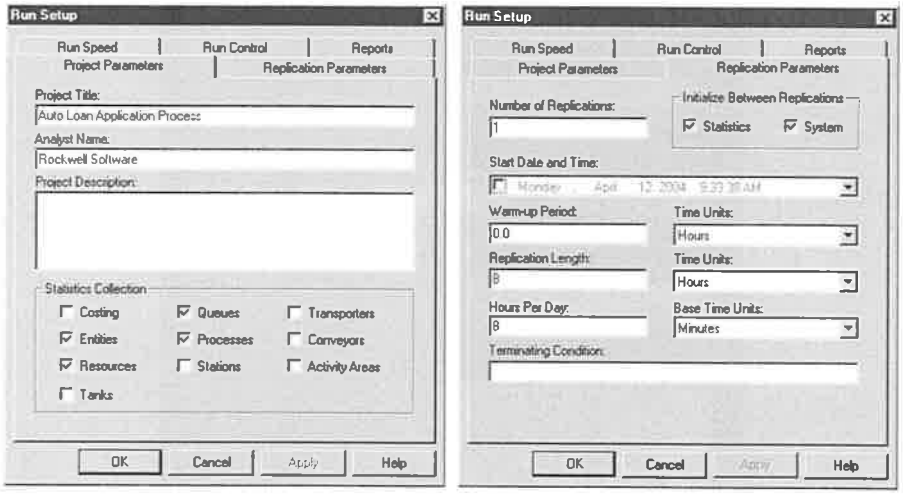
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	r Loan Officer	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑

Running the Model

Run Setup

Defines the parameters of the simulation run, including the simulation run length (and units), number of hours/day, warm-up time, and number of simulation replications. It allows the user to turn off various groups of statistics to the summary reports, including process, entity, resource, and queue statistics.

The run setup can be accessed by choosing the **Run > Setup** option from the menu at the top of the screen.



Number of Replications	1
Replication Length	8 hours
Hours per Day	8 hours

When you are ready to begin simulating your model, use the Go command ▶ (Run menu) to start the simulation run. Arena will first check your model to ensure that you've defined a valid model, and then it will launch the simulation.

When you begin running a simulation model, you enter a special mode called a run session. Once initiated, you remain in the run session until you explicitly terminate it by selecting the End command ■.

Reports

There are multiple reports automatically generated for every model. These reports are separated into different categories, each one detailing a certain type of statistic. Depending on the system defined in the model, some of these categories may be empty and the corresponding report will be omitted.



Reports — Category Overview

Displays entity, process, queue, and resource information summarized for all replications. It also displays any user-assigned variables or items defined in the Record module.

Contains the categories:

Entity Reports and Process Reports

- Cycle Time
- NVA Cost, NVA Time
- Total Cost
- VA Cost, VA Time
- Wait Cost, Wait Time
- WIP

Queue Reports

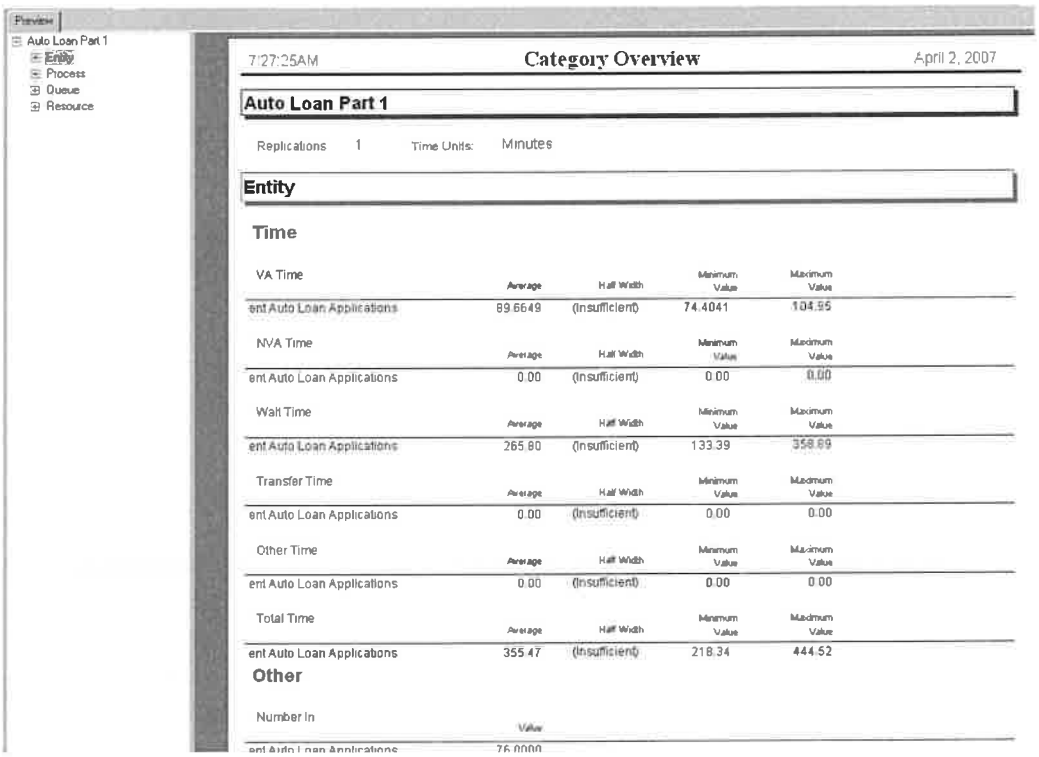
- Number Waiting
- Wait Cost, Wait Time
- Resource Reports
- Number Busy
- Number Scheduled
- Utilization

Resource Reports

- Number Busy
- Number Scheduled
- Utilization

User-Specified Reports

Category Overview Report from Auto Loan Part 1



Preview

- Auto Loan Part 1
  - Entity
  - Process
  - Queue
  - Resource

7:27:25AM **Category Overview** April 2, 2007

Auto Loan Part 1

Replications 1 Time Units: Minutes

Entity

Time

VA Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applications	89.6649	(Insufficient)	74.4041	104.95

NVA Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applications	0.00	(Insufficient)	0.00	0.00

Wait Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applications	265.80	(Insufficient)	133.39	358.69

Transfer Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applications	0.00	(Insufficient)	0.00	0.00

Other Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applications	0.00	(Insufficient)	0.00	0.00

Total Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applicabons	355.47	(Insufficient)	218.34	444.52

Other

Number In	Value
ent Auto Loan Applications	76.0000

Topics covered in this section:

Concepts

- Creating and using user-specified and Arena internal variables
- Animation of variables (Scoreboard)
- Conditional and probability-based decisions

Arena Constructs

- Variable spreadsheet module
- Decision logic module
- Assign logic module
- Animate toolbar (Variable)

Concepts and Terminology

Variables

Variables represent a set of global values that can be altered or queried anywhere within the model. Initial values can be set by altering the variables module, or by reading in data from outside sources. If initial values are not set, the variables have a default value of 0. Arena has two types of variables: user-defined and system.

User variables are defined (named) by the modeler and can be changed during the simulation run (within the model or interactively).

Examples:

arrival rate
current inventory
number of patients registered
and so on.

System variables are predefined characteristics of model components that provide the state of the component.

Examples:

number of entities waiting in a queue; NQ(queue name)

current value of a counter; NC(counter name)

Animation Concepts and Terminology

Pictures

Pictures are used to animate entities. As an entity is processed through the model, assigning a new picture name can change its picture.

Multiple drawings with the same picture name may now be defined. During a model run, Arena will randomly select one of the drawings to animate an entity when the picture name is assigned.

Resources

Resource pictures can change to show the status of a resource. The four default states (idle, busy, inactive, and failed) can be augmented by user-defined states.

Queues

Queues are animated by a line or series of points that define the location for entity pictures to be drawn.

When an entity enters a queue, Arena provides the option to animate the entity moving smoothly along the drawn queue line to its rank position in the queue. The speed of a queue's arrival animation is configurable.

Variables (Scoreboard)

Variables and expressions are animated by a display (scoreboard) that shows the current value of the variables or expressions. This value is updated as variable or expression values change.

Auto Loan Part 2 (Auto Loan Part 2.doe)

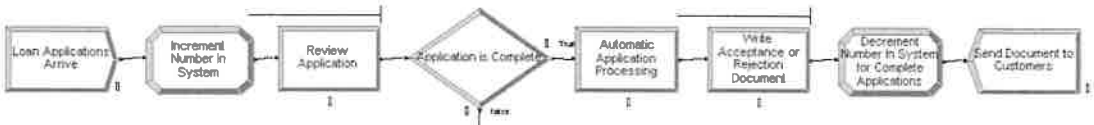
The following section demonstrates additional basic tools used to create simulations with the Basic Process panel. This example simulates the previous car loan application process in greater detail.

After some review of the system, the loan officers determined that 8% of the applications they review are incomplete. These incomplete applications are sent back to the applicant.

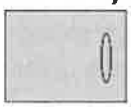
There was also some question as to the actual number of applications in process throughout the simulation period. This value will be captured in a user-defined variable.

In addition, use the Animation toolbar to animate the number of applications in process.

Run this simulation for one 8-hour day.



Number in System



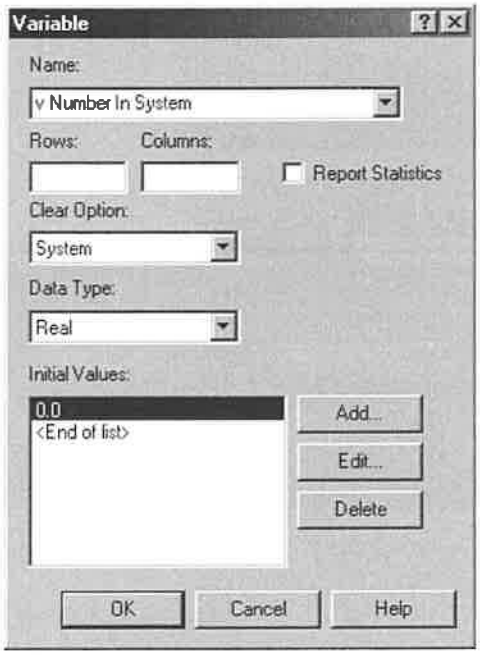
Data Modules

Variable Module



Variable

Defines the global variables used in the simulation. Variables can be single elements, 1-D arrays, or even 2-D arrays. Initial variable values may be defined; if values are not defined, a default of 0 is set.



Variable - Basic Process

	Name	Rows	Columns	Data Type	Clear Option	Initial Values	Report Statistics
1	v Number In System			Real	System	1 rows	☐

Double-click here to add a new row

Flowchart Modules

Assign Module

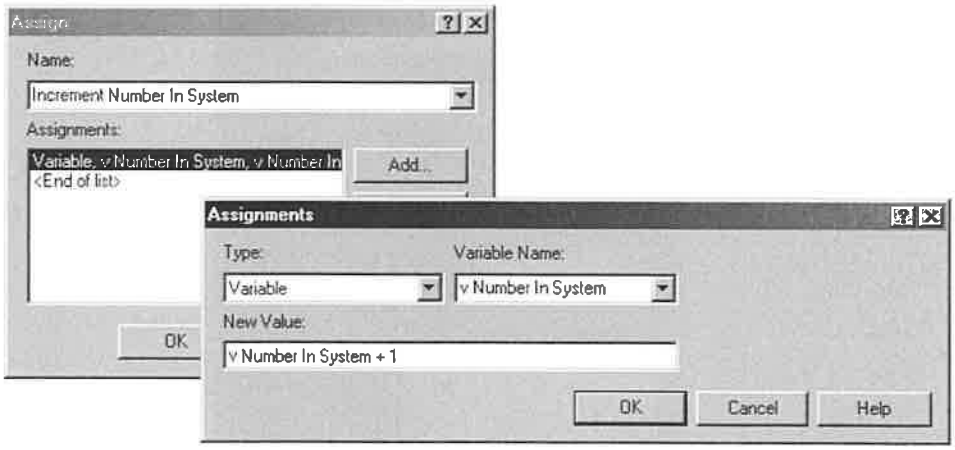


Assign

This module is used for assigning new values to variables, entity attributes, entity types, entity pictures, or other system variables.

For now we will use the Assign only for incrementing a variable, but later we'll use the Assign module to alter the values of other quantities and attributes.

Increment Number in System — Use the Assign module



Name	Increment Number in System
Type	Variable
Variable Name	v Number in System
New Value	v Number In System + 1

Decide Module



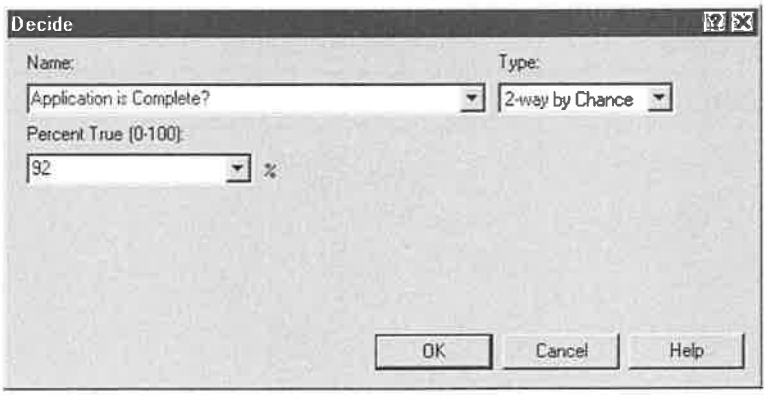
Decide

Allows for decision-making processes in the system. It includes options to make decisions based on a condition or based on a probability.

The decisions can be one of the following four types:

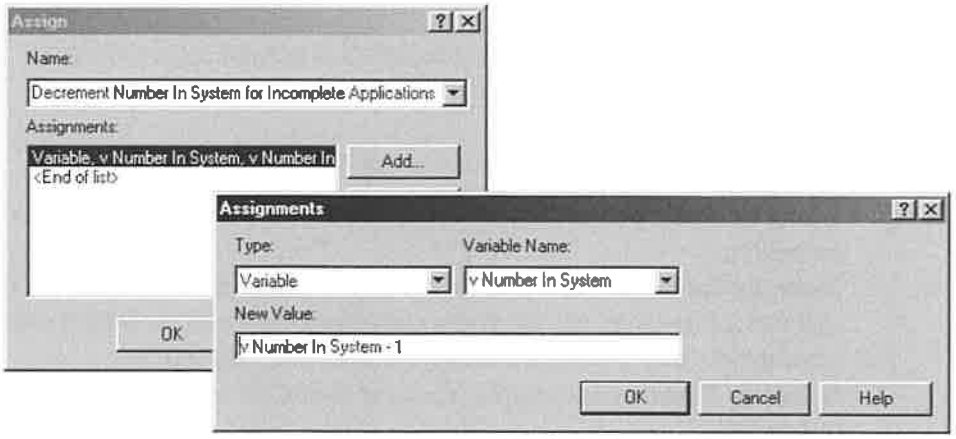
2-way by Chance – Entity can go to one of two possible locations based on a probability.
2-way by Condition – Entity can go to one of two possible locations based on a user-defined condition. The conditions can include the number of entities waiting in a queue, the value of a global variable, the value of an attribute, etc.
N-way by Chance – Entity can go to one of N possible locations based on a probability.
N-way by Condition – Entity can go to one of N possible locations based on user-defined conditions.

Application is Complete — Use the Decide module



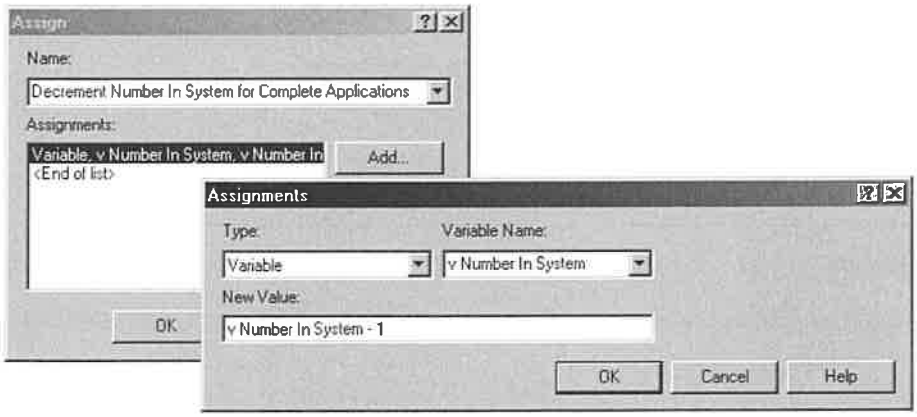
Name	Application is Complete?
Type	2-way by Chance
Percent True (0-100)	92 %

Decrement Number in System for Incomplete Applications — Use the Assign module



Name	Decrement Number In System for Incomplete Applications
Type	Variable
Variable Name	v Number In System
New Value	v Number In System - 1

Decrement Number in System for Complete Applications — Use the Assign module



Name	Decrement Number In System for Complete Applications
Type	Variable
Variable Name	v Number In System
New Value	v Number In System - 1

Animation Objects

The Animate toolbar provides the only interface to Arena's animation objects (they are not available on any of the Arena menus).



There are three types of commonly used animation objects:

Queues

Displays entities that are waiting for an event to occur (e.g., a resource becoming available).

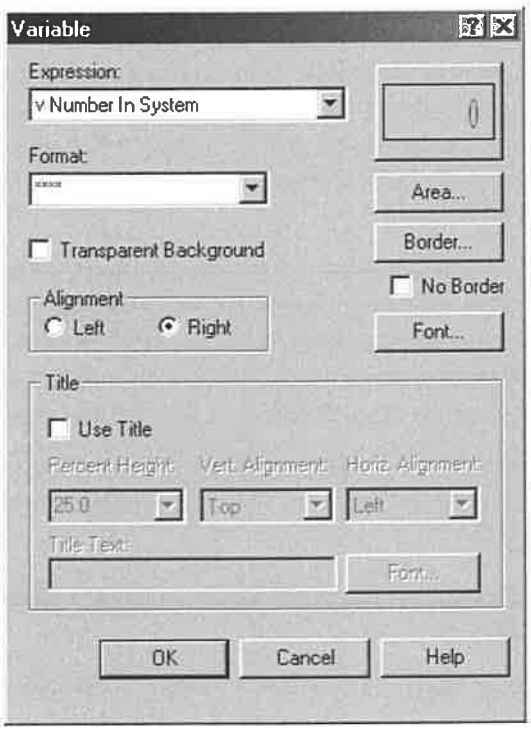
Resources

A resource can be animated with a unique picture associated with one of its states: Idle, Busy, Inactive, Failed, or any user-defined states. During the run, the resource picture changes based on the current state of the resource.

Status Displays

Clock	Date	Variable
Level	Histogram	Plot

Animate the Number in System — Use the Variable Animation Object



Expression	v Number In System

Basic Process Workshop (Basic Process Workshop.doe)

Requisition Fulfillment Process

The purchasing department of the HJW Corporation receives approximately 48 new purchase requests a day (about 1 every 10 minutes). As requests arrive, an intern reviews the purchase amount of each request and sends them to the appropriate location based upon the purchase amount. This review process takes anywhere from 20 to 60 seconds, but usually requires 30 seconds.

Approximately 25% of the requests are for items costing more than $2000. These requests are sent to the appropriate department head for approval before they proceed to purchasing. This process can take up to 1 work day or as short as 2 hours; however, most take 4 hours. Only 50% of these requests are approved and sent on to a purchasing agent. Requests less than or equal to $2000 are sent directly to a purchasing agent.

Once a request is assigned to a purchasing agent, the agent fills out the purchase order form and phones the order to the vendor. This process takes anywhere from 25-35 minutes to complete.

Once the agent is finished, the intern will file the completed purchase requests. This task requires 2-3 minutes.

The corporation has 3 purchasing agents and 1 intern working full time in the purchasing department.

The base time units should be in minutes, and there should be 8 hours in a day.

Run the simulation for 5 days.

Examine the Utilization of Resources and the Queue Waiting Times.

Model the Requisition Fulfillment activities using the Basic Process modules.

ADDITIONAL EXERCISES:

- Capture the Number In System and animate this User-Defined Variable.

4: Resource Sets and Schedules

Topics covered in this section:

Concepts

- Limiting resource availability (Schedules)
- Cross utilization of resources (Sets)
- User-defined entity arrivals

Arena Constructs

- Resource spreadsheet module
- Schedule spreadsheet module
- Set spreadsheet module
- Create logic module

Seize, Delay, and Release Revisited

Sometimes when using a resource, you want an entity to seize a resource and retain control of it for several processes before releasing it. The Process module has a function that allows an entity to seize a resource and hold onto it for several processes.

The "Seize Delay Release" and "Delay" options within the Process module were discussed previously.

With the "Seize Delay Release" command, an entity seized a resource, performed one process, and then released the resource.

The "Delay" option delayed an entity within the system without requiring a resource.

However, if you wish to have an entity maintain control of a resource, use the "Seize Delay" option within the Process module. With this command, an entity will seize a resource and delay. Once the delay is over, the entity will move to the next module while still maintaining control of the resource. You can now move the entity through multiple modules while maintaining control of the resource.

Once you are ready to release the resource, use another Process module with the "Delay Release" option.

Entities are not limited to gaining control of only one resource at a time. They have the ability to seize multiple resources at one instance, and/or seizing resources over multiple processes. Likewise, these resources can be released in a similar fashion (releasing all of the resources with one "Delay Release" option, or by releasing after multiple "Delay Release" options). There are almost limitless combinations in which the three steps ("Seize," "Delay," and "Release") can be used. Their individual use will be discussed in greater detail in the Advanced Process portion of the training.

If you want to seize or release a resource and not have a delay, set the delay time to zero.

Remember that if an entity seizes a resource, the entity must release the resource before it exits the system.

Resource Schedules

Sometimes resources are not always available throughout the length of the simulation run. In order to accommodate the changes in resource availability, Arena allows the user to create schedules for any resource.

A schedule is a time-based pattern of changes in resource capacity.

A schedule is specified by entering repeating pairs of resource capacity and the time duration for the capacity. The capacity and the duration can be entered as random quantities or constants.

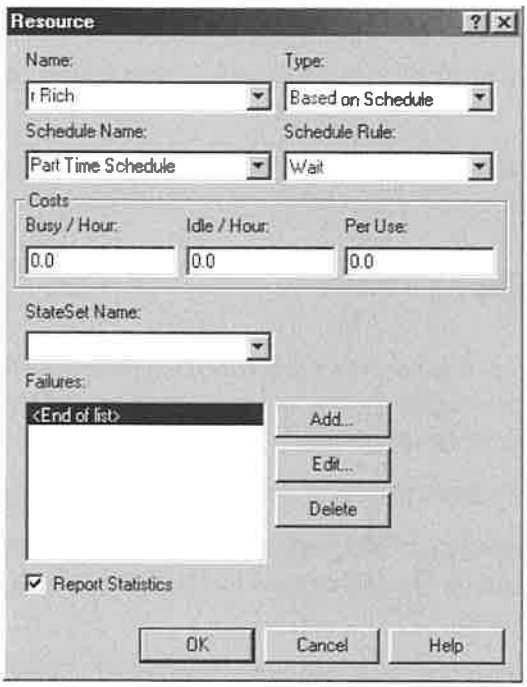
Resource schedule information is reported as the number of available resource units statistic.

Arena has three options for handling the scheduled change in resource capacity: Preempt, Wait, and Ignore.

Each resource can only have one schedule.

Resource Module Revisited

Within the Resource module, you may specify that the given capacity of a resource is constant, or you may specify that the resource follow a schedule.



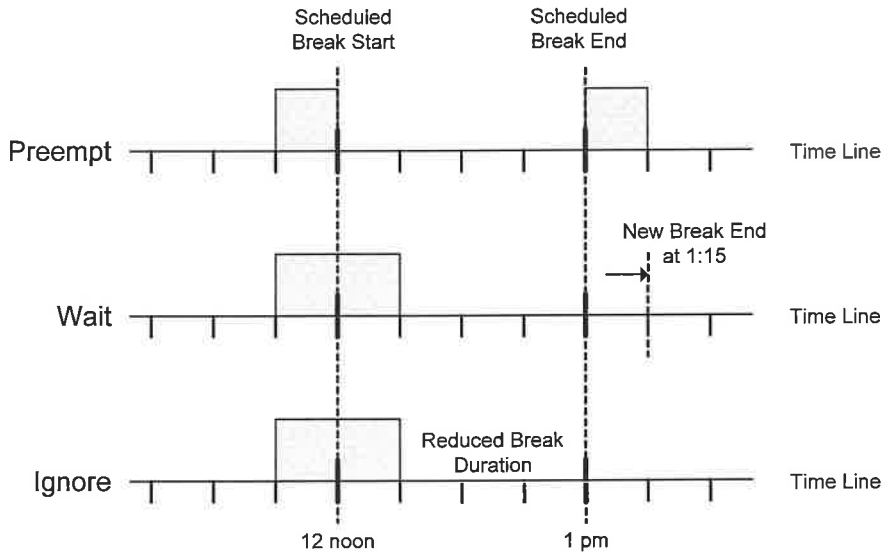
If the resource capacity is based on a schedule, the schedule will be defined in the Schedules module.

Preempt, Wait, and Ignore Defined

Preempt: Capacity change takes place immediately. Processing of the entity is completed when the resource becomes available.

Wait: Processing of the entity is completed before the capacity change takes place. The next scheduled capacity change will be offset.

Ignore: Capacity change takes place immediately, but the resource continues processing. This impacts the duration of the capacity change.



Schedule Module

This data module must be used to define an operating schedule for a resource.

Schedule - Basic Process

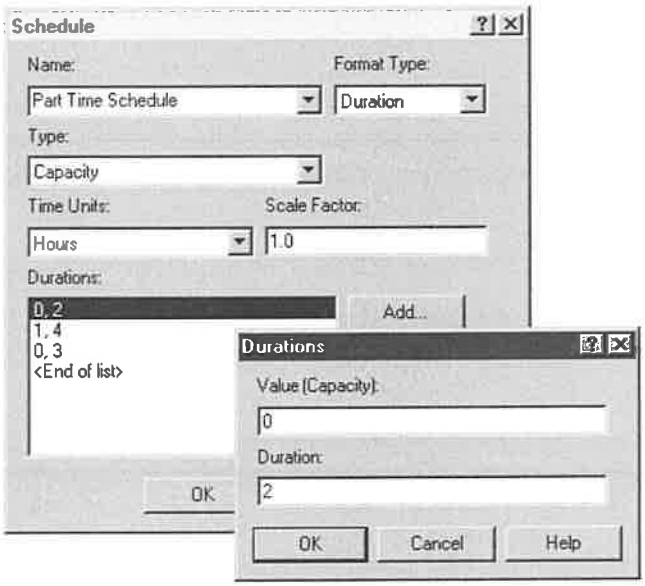
	Name	Format Type	Type	Time Units	Scale Factor	Durations
1	Driver Arrival Schedule	Duration	Arrival	Hours	1.0	9 rows
2	Part Time Schedule	Duration	Capacity	Hours	1.0	3 rows
3	Full Time Schedule	Duration	Capacity	Hours	1.0	3 rows
4	Manager Schedule	Duration	Capacity	Hours	1.0	3 rows

Double-click here to add a new row.

The Schedule module can be created and edited in two ways: the Graphical Schedule Editor, or editing the dialog.

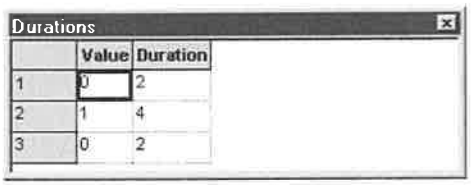
Schedule Editor via Editing the Dialog

The dialog for the Schedule editor is activated by selecting the schedule to be edited, right-clicking the mouse, and selecting "Edit via Dialog."



Schedule Editor via Editing the Spreadsheet

The spreadsheet for the Schedule editor is activated by selecting the schedule to be edited, right-clicking the mouse, and selecting "Edit via Spreadsheet."



Sets

Sometimes different resources can be utilized to complete the same tasks, but these resources might follow different schedules or be utilized based on a priority scheme. In order to accommodate this problem, a resource set should be used.

Sets group similar elements so that they can be referenced via a common name and set index.

Typical sets would contain groups of resources, queues, stations, pictures, counters, tallies, or expressions. Individual sets can only contain elements of the same type; mixed elements cannot be contained in a set (that is, a single set cannot contain both queues and resources).

Objects in a set can be referenced directly or via the set name and index.

Members of a resource set can be selected using Set Name(set rule). Referencing the set using the set name and a set rule will choose the members in the set according to the rule.

There are several resource set selection rules that may be specified: Cyclical, Random, Preferred Order, Specific Member, Largest Remaining Capacity, and Smallest Number Busy.

When using the Preferred Order selection rule, the order of the resources specified in the Resource Set module may be important.

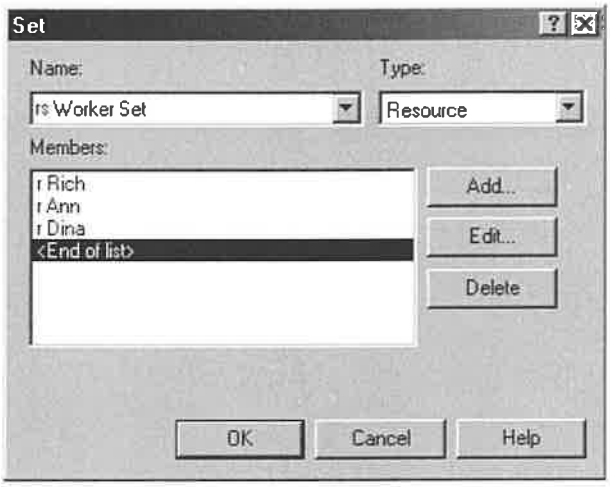
Example: A resource set called First Shift contains Mary, Bob, Jane, and Dave. Mary could be referenced directly by quoting the element name "Mary." By referencing via the set name and an index, First Shift (2) would return the resource Bob.

Elements can be members of more than one set, but all members of the set must be of the same type.

Why Use Sets?

- Individually animate resources
- Individual statistics
- Members can follow different schedules
- Cross-utilized resources
- Ability to create different process times based on skills, age, etc.

The Sets module allows us to form sets of resources. These resource sets are referenced in the Process module when specifying the resource required for that activity.



Arrival Schedules

Sometimes entity arrival rates are not always constant throughout the length of the simulation run. In order to accommodate the changes in entity arrival rates, Arena allows the user to define arrival schedules for any Create module.

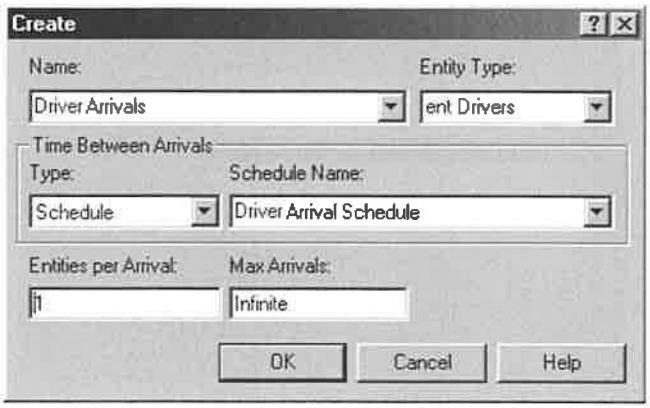
An arrival schedule is a time-based pattern of changes in the arrival rate of entities for a given Create module expressed in number of entities per hour.

An arrival schedule is specified by entering repeating pairs of entity arrival rate and the time duration for the arrival rate. The arrival rate and the duration can be entered as random quantities or constants.

Each Create module can have only one schedule.

Create Module Revisited

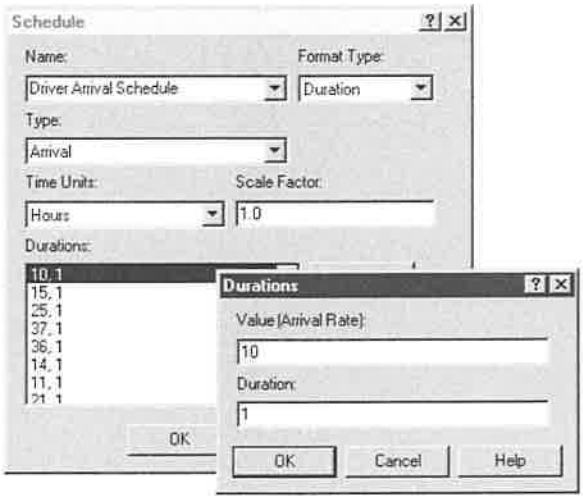
The Type field allows the user to define one of four arrival types: Random, Schedule, Constant, or Expression.



Schedule Module Revisited

When using arrival type schedules, the value specified is the number of arrivals per hour.

Edited via Dialog



DMV Photo License Center — Part 1 (DMV Photo License Center – Part 1.doe)

At the DMV photo license center, the current "rule of thumb" is to get the drivers in and out as fast as possible. Therefore, when drivers come to get their photo ID, the following activities occur: Drivers go to a large counter, pick up a blank form, and fill it out. (We'll assume the counter is large enough that any number of drivers can obtain and complete the form at one time.) The drivers then wait in line for an available photo clerk.

The process goes as follows: The clerk reviews the form with each driver, takes the driver's photo, captures the driver's signature, and prints the completed license. (Note: A computer is needed for both the signature capture and for printing the license.) The DMV manager knows that it normally takes 3 minutes (some as short as 2 minutes, others as long as 4 minutes) to review the form and take a driver's photo, and anywhere between 30 and 60 seconds to capture the driver's signature electronically. It then takes exactly 45 seconds to print and seal the new photo driver's license.

The drivers are serviced by: 1 part-time clerk (Rich) who works daily from 10 a.m. to 2 p.m., 1 full-time clerk (Ann) who works daily from 8 a.m. to 5 p.m. with an hour break between 10:30 a.m. and 11:30 a.m., and the site manager (Dina) who only helps when everyone else is busy and there are more drivers to be served. The manager works daily like Ann, but gets her break from 1:15 p.m. to 2:15 p.m. Due to budgetary constraints, there are only 2 computers available at the facility. (Don't worry about using sets for this resource.) The computers are used to capture the driver's signature and to print the license.

DMV Photo License Center — Part 1 (cont.)

Most of the drivers arrive during their lunch break between 11 a.m. and 1 p.m. Before and after the lunch rush, drivers trickle in and out. The chart below defines the average number of drivers who arrive by hour over the course of the 9-hour day.

Hour	*Avg. # of Drivers*
8 a.m. – 9 a.m.	10
9 a.m. – 10 a.m.	15
10 a.m. – 11 a.m.	25
11 a.m. – Noon	37
Noon – 1 p.m.	36
1 p.m. – 2 p.m.	14
2 p.m. – 3 p.m.	11
3 p.m. – 4 p.m.	21
4 p.m. – 5 p.m.	24

Run the simulation for one 9-hour day. The base time units should be minutes.

Modeling Approach

Driver Arrivals: Use a Create module that follows an arrival schedule.

Review Form and Take Photo: Seize an operator and delay.

Capture Signature Electronically: Seize a computer and Delay for the signature time while maintaining control of the camera operator.

Print and Seal the License: Delay for the printing and sealing processes, then release the correct operator and computer.

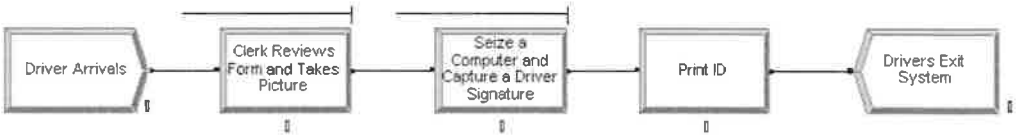
Worker Schedules: Create a schedule for each operator.

Worker Priorities: Create a resource set and seize the operators using the preferred order rule.

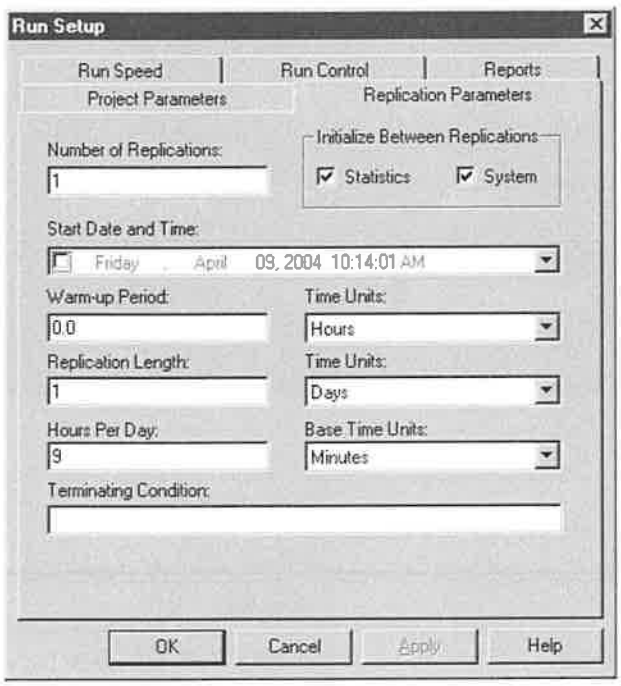
Driver Arrival Schedule: Create an arrival schedule for the Create module.

DMV Photo License Center — Part 1

DMV License Photo Center - Part 1.doe

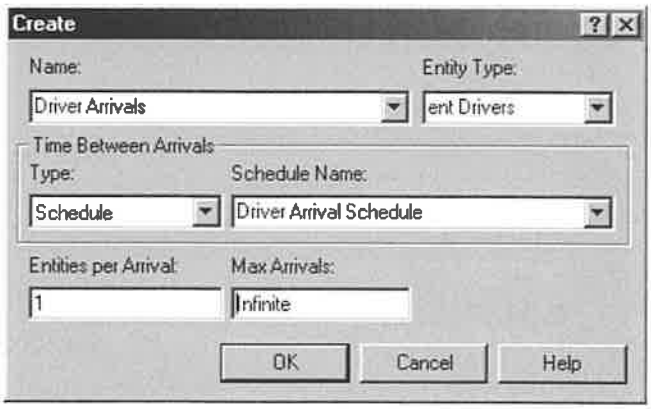


Define Replication Parameters — Use **Run > Setup**



Number of Replications	1
Warm-up Period	0 hours
Replication Length	1 days
Hours per Day	9
Base Time Units	Minutes

Driver Arrival — Use the Create module



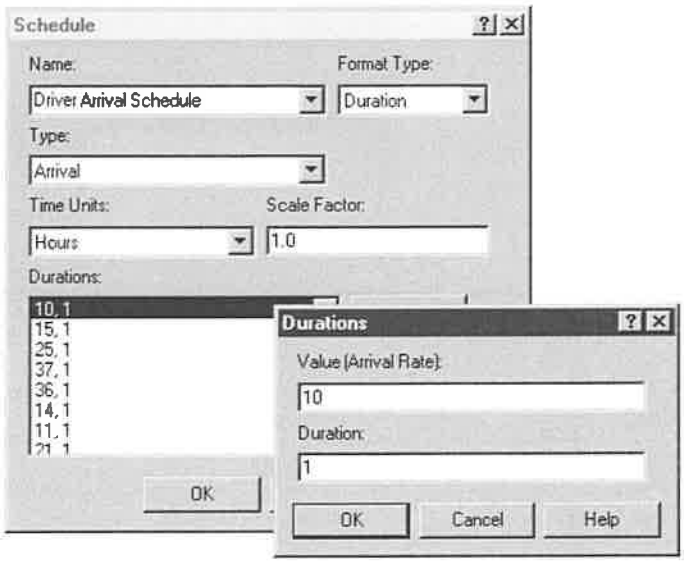
Name	Driver Arrivals
Entity Type	ent Drivers
Type	Schedule
Schedule Name	Driver Arrival Schedule
Assign initial picture in Entity Spreadsheet	

Define Driver Arrivals — Use the Schedule module

Schedule - Basic Process

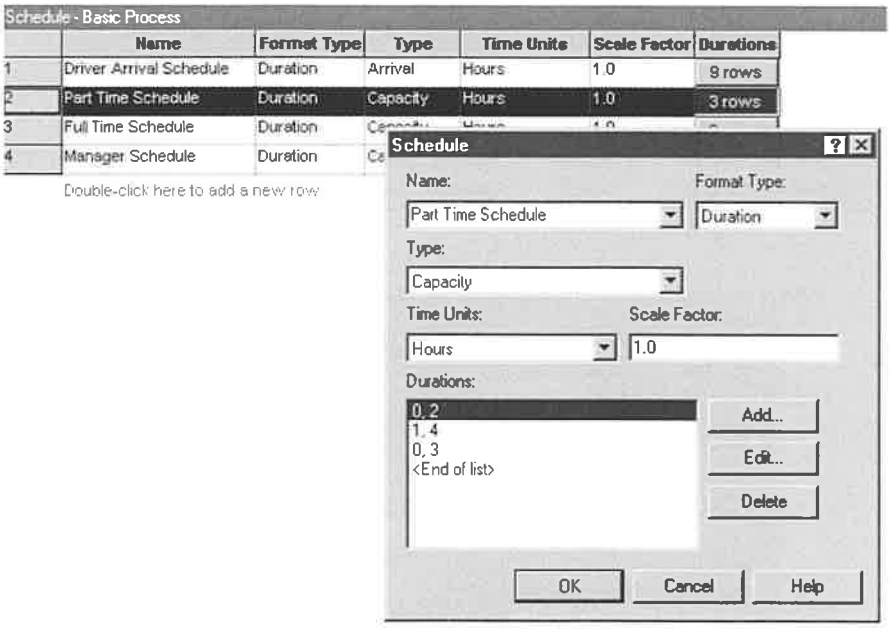
	Name	Format Type	Type	Time Units	Scale Factor	Durations
1	Driver Arrival Schedule	Duration	Arrival	Hours	1.0	9 rows
2	Part Time Schedule	Duration	Capacity	Hours	1.0	3 rows
3	Full Time Schedule	Duration	Capacity	Hours	1.0	3 rows
4	Manager Schedule	Duration	Capacity	Hours	1.0	3 rows

Double-click here to add a new row



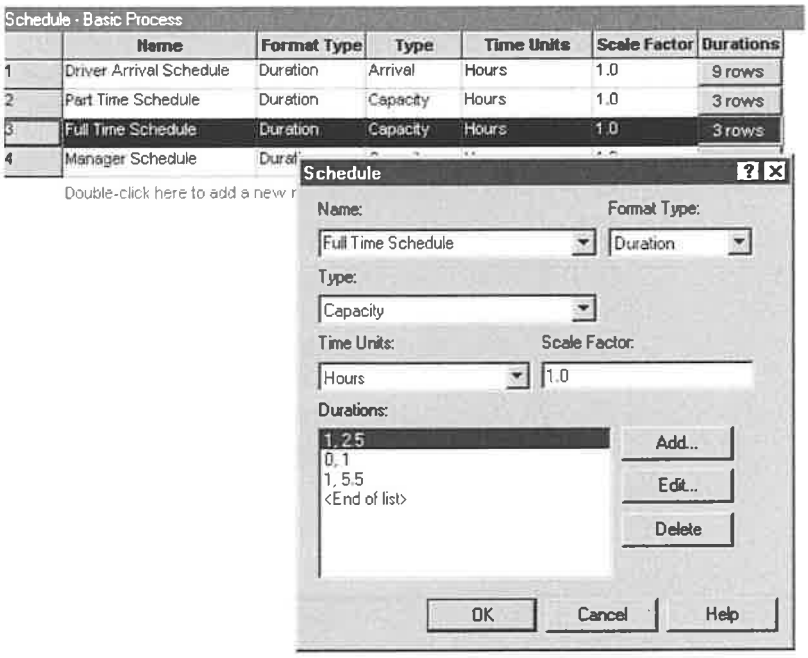
Name	Driver Arrival Schedule
Type	Arrival
Time Units	Hours
Scale Factor	1
Durations	Insert the correct values and durations for the 9-hour day.

Define Part Time Schedule — Use the Schedule module



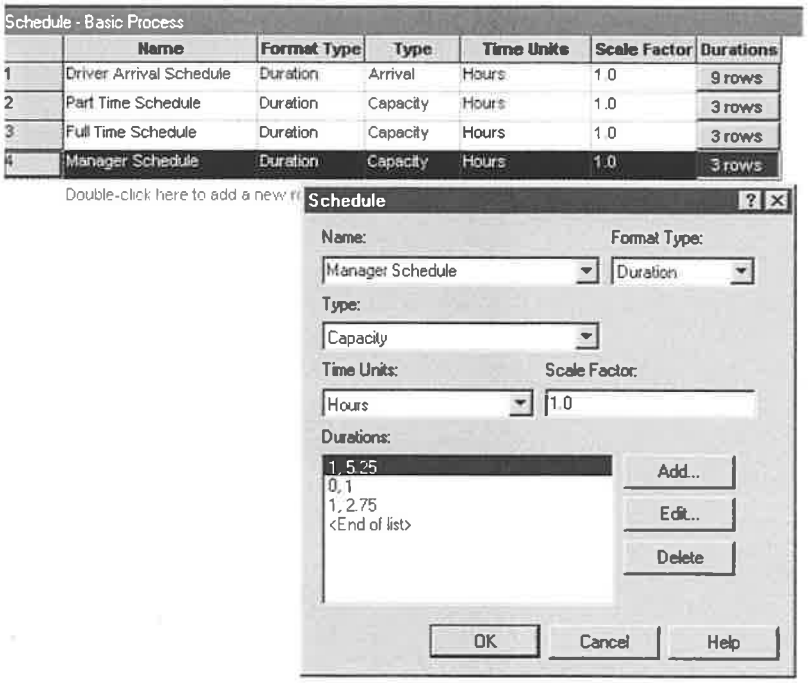
Name	Part Time Schedule
Type	Capacity
Time Units	Hours
Durations	Insert the correct values and durations for the 9-hour day.

Define Full Time Schedule — Use the Schedule module



Name	Full Time Schedule
Type	Capacity
Time Units	Hours
Durations	Insert the correct values and durations for the 9-hour day.

Define Manager's Schedule — Use the Schedule module



Name	Manager Schedule
Type	Capacity
Time Units	Hours
Durations	Insert the correct values and durations for the 9-hour day.

Form the Resource Set — Use the Set module



Name	rs Worker Set
Type	Resource
Members	r Rich, r Ann, r Dina

Remember to list the resources in the order of preference.

Assign Schedules or Capacity to Resources — Use the Resources module



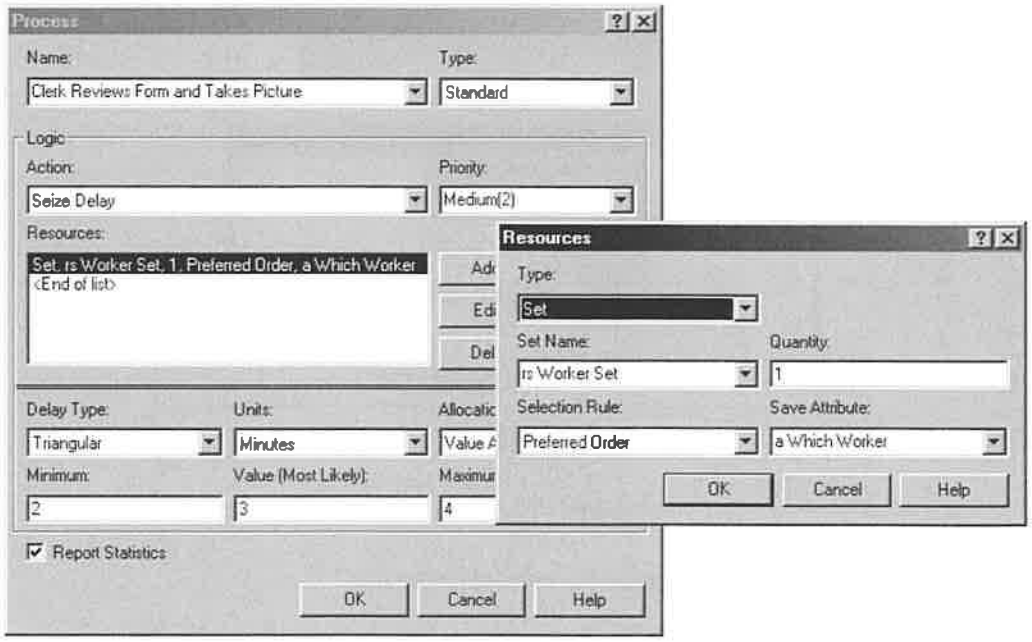
Name	r Rich
Type	Based on Schedule
Schedule Name	Part Time Schedule

Assign the schedules for Ann and Dina following the steps above.

Next assign the capacity of the Computer resource.

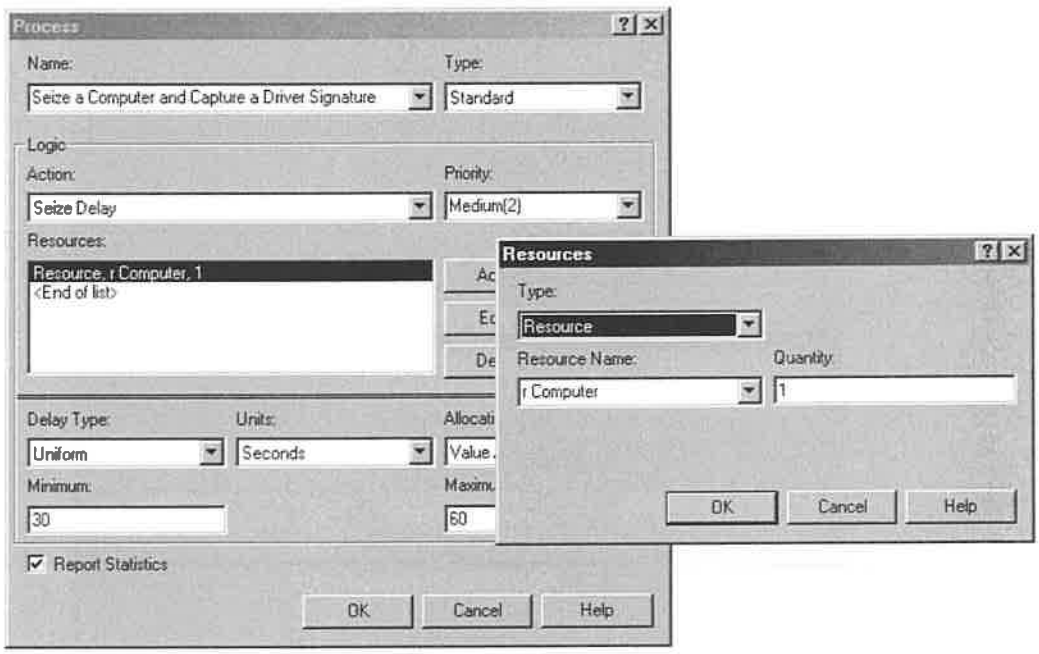
Name	r Computer
Type	Fixed Capacity
Capacity	2

Seize a Worker to Review Form and Take Photo — Seize from a resource set using a Preferred Order rule



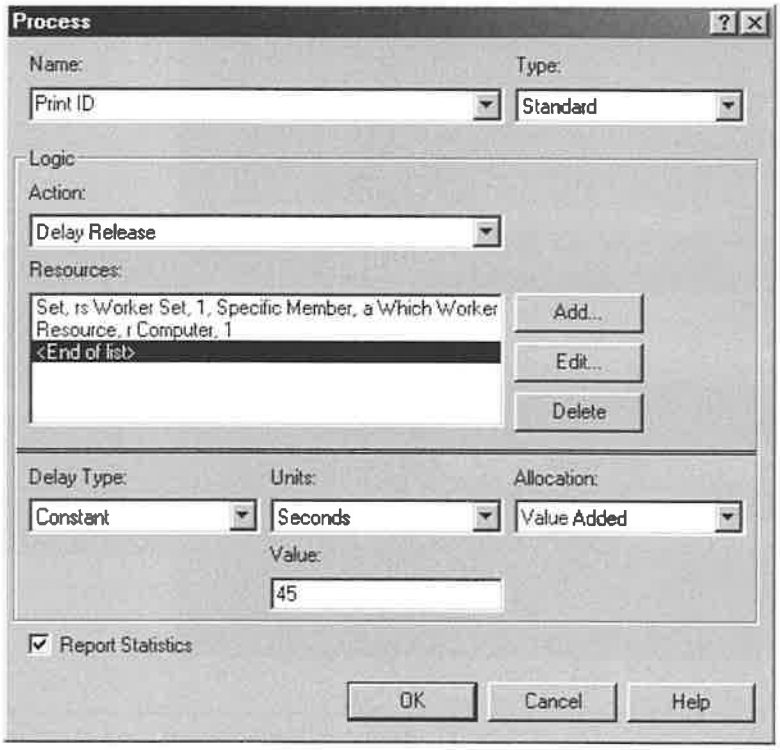
Name		Clerk Reviews Form and Takes Picture
Action		Seize Delay
Resources	Type	Set
	Set Name	rs Worker Set
	Quantity	1
	Selection Rule	Preferred Order
	Save Attribute	a Which Worker
Delay Time		Triangular(2,3,4) minutes

Seize a Computer — Seize one of the computers and delay to receive the driver's signature.



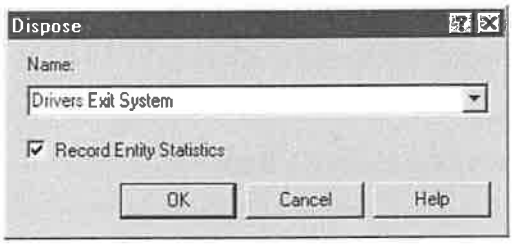
Name		Seize a Computer and Capture a Driver Signature
Action		Seize Delay
Resources	Type	Resource
	Resource Name	r Computer
	Quantity	1
Delay Time		Uniform(30,60) Seconds

Print Photo ID — Use the Process module to delay for printing the ID and releasing the worker and computer seized.



Name	Print ID	
Action	Delay Release	
Resources	Type	Set
	Set Name	rs Worker Set
	Quantity	1
	Selection Rule	Specific Member
	Set Index	WhichWorker
	r Computer	
Delay Time	45 seconds	

Drivers Exit — Use the Dispose module



Name	Drivers Exit System
Record Entity Statistics	*check*

Save and run the model.

Topics covered in this section:

Concepts

- Entity, Queue, and Resource animation

Arena Constructs

- Animate toolbar (Resource, Clock)

Animation Concepts and Terminology

Pictures

Pictures are used to animate entities. As an entity is processed through the model, assigning a new picture name can change its picture.

Multiple drawings with the same picture name may now be defined. During a model run, Arena will randomly select one of the drawings to animate an entity when the picture name is assigned.

Resources

Resource pictures can change to show the status of a resource. The four default states (idle, busy, inactive, and failed) can be augmented by user-defined states.

Queues

Queues are animated by a line or series of points that define the location for entity pictures to be drawn.

When an entity enters a queue, Arena provides the option to animate the entity moving smoothly along the drawn queue line to its rank position in the queue. The speed of a queue's arrival animation is configurable.

Variables (Scoreboard)

Variables and expressions are animated by a display (scoreboard) that shows the current value of the variables or expressions. This value is updated as variable or expression values change.

DMV Photo License Center — Part 2 (DMV Photo License Center – Part 2.doe)

After having reviewed the original Photo License Center model, the DMV is interested in developing a more detailed representation of their system. Along with increased detail, the DMV would also like to see some basic animation, enabling a better visual representation of the photo license center.

At the present time, no retakes are allowed. After a study, it was discovered that about 25% of the drivers are unhappy with their "mug shots" and ask whether their photo can be re-taken.

After much discussion with the DMV employees, the DMV will consider allowing one photo re-take. They estimate that a retake will take anywhere between 20 and 40 seconds. After a retake, the driver will continue on with the process as it was previously performed (Get Signature and Print ID).

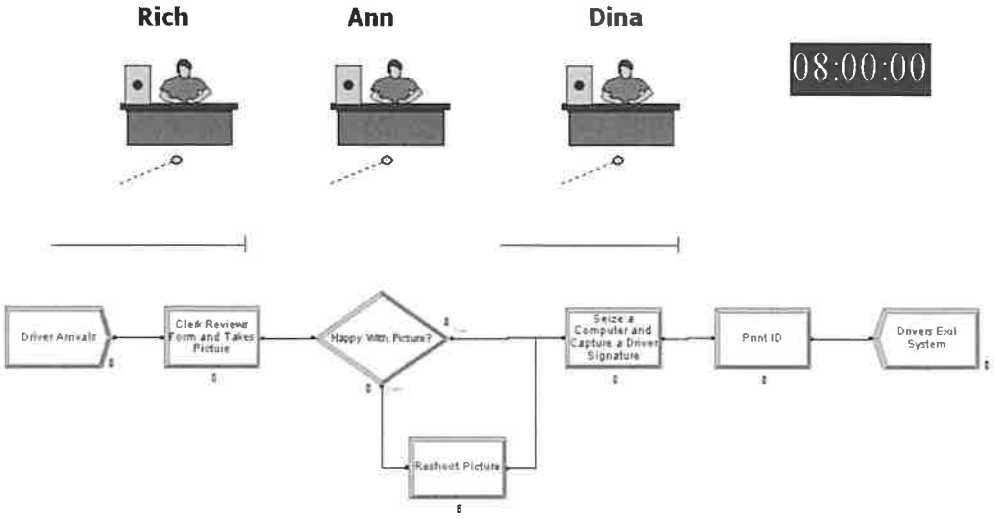
To aid in giving a better visual representation of the system, the DMV would like animation for the worker resources. In addition, the DMV would also like to see two different picture representations of the drivers and smooth movement into the queues. Also, add a clock to the system.

Once again, run the simulation for one 9-hour day.

Save the model under a new file name, and run the new model. Compare the drivers' time in system to the original model.

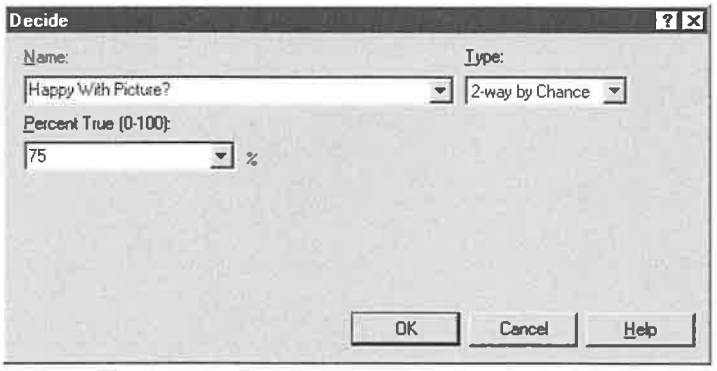
DMV Photo License Center — Part 2

DMV Photo License Center - Part 2.doe



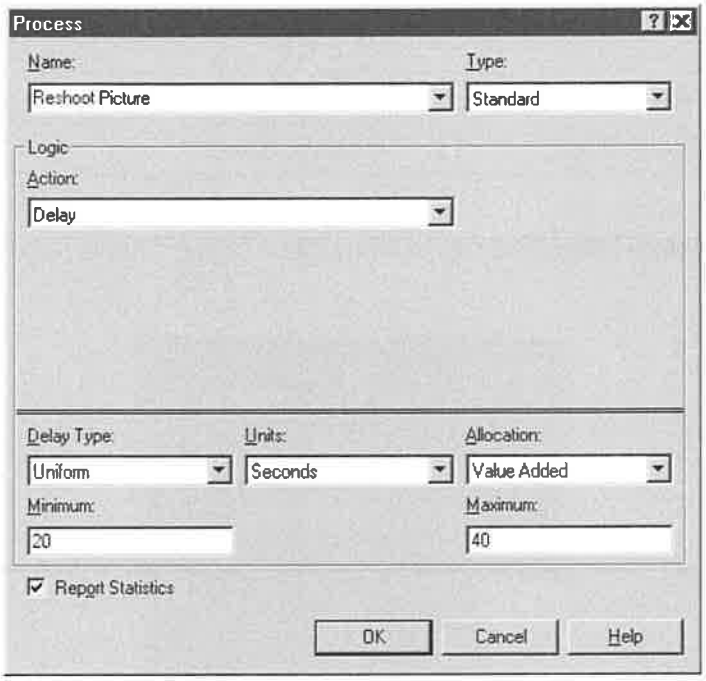
Using the previously built "as is" Photo License Center Model – Part 1, add the option for a retake and animation of worker resources and changing driver entity pictures.

Allow a Retake — Use the Decide module



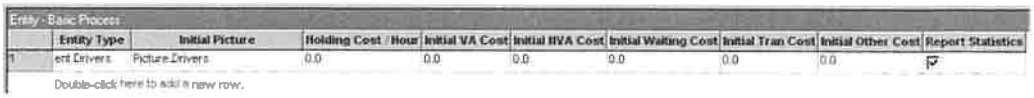
Name	Happy With Picture?
Type	2-way by Chance
Percent True	75

Delay for Retake — Use the Process module



Name	Reshoot Picture
Action	Delay
Delay	Uniform (20,40) seconds

Assign Picture to Driver entities — Use the Entity module

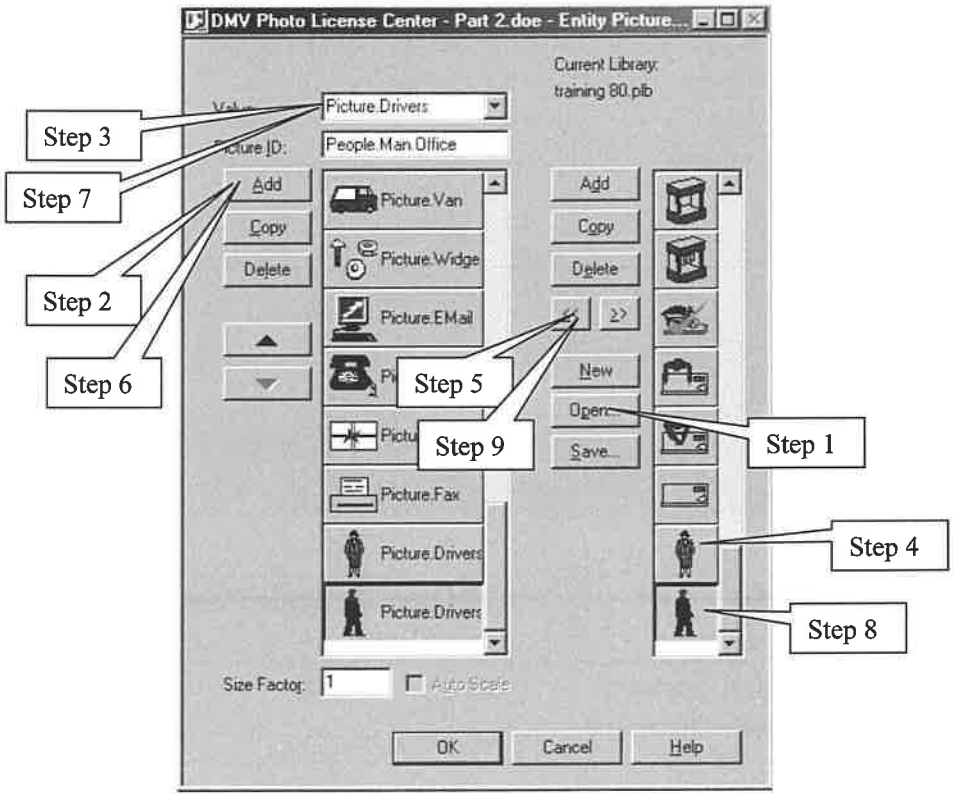


Entity Type	ent Drivers
Initial Picture	Picture.Drivers

Create Custom Entity Pictures — Select **Edit > Entity Pictures** from the menu bar



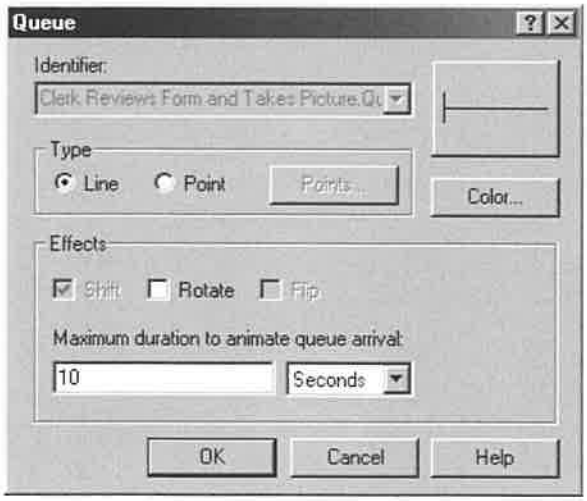
Creating Custom Entity Pictures



Step 1	Open the "training 80.plb" picture library
Step 2	"Add" a new entity picture
Step 3	In the Value field, Name the new picture: "Picture.Drivers"
Step 4	Select the picture from the picture library that you would like to use
Step 5	Transfer the picture from the library to the new picture
Step 6 - 9	Repeat Steps 2 through 5

NOTE: The name of the entity picture does not have to contain the prefix "Picture."

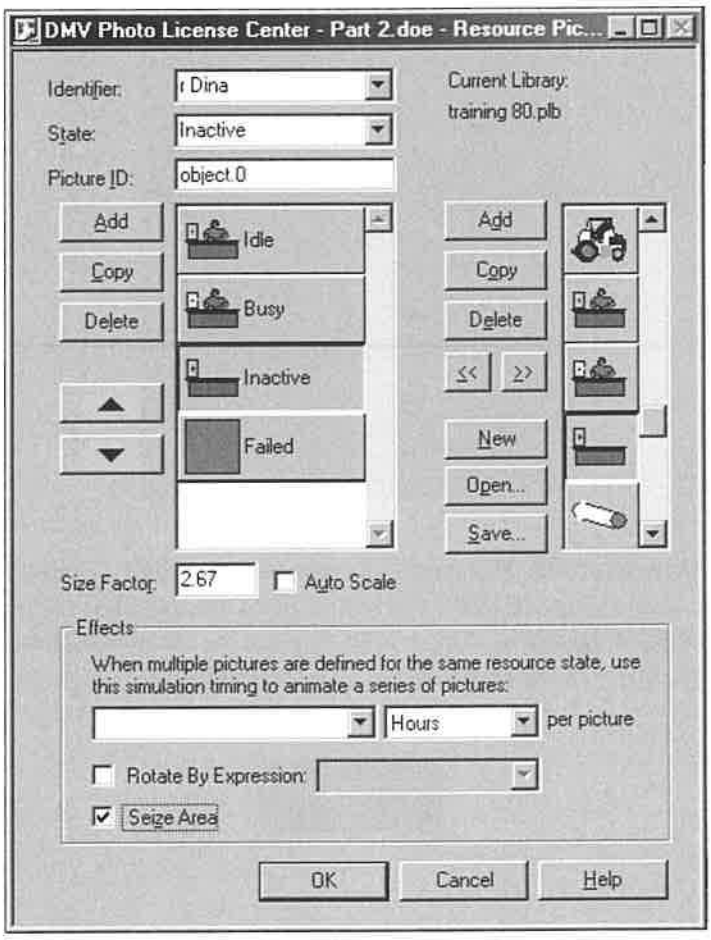
Animate Travel in Queue — Use the Queue button from the Animate toolbar



Identifier	Clerk Reviews Form and Takes Picture.Queue
Maximum Duration to Animate Queue Arrival	10 Seconds

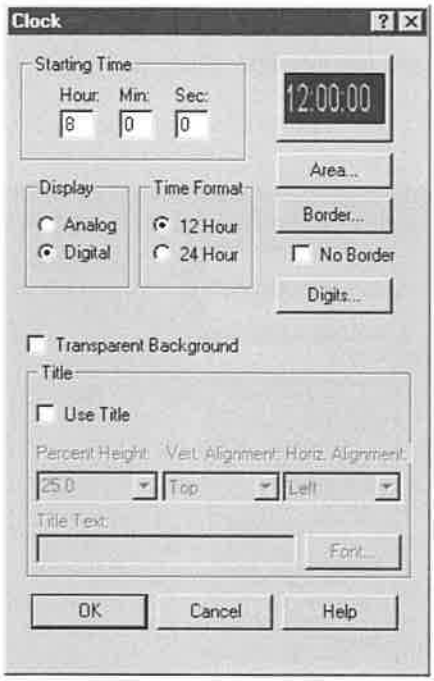
Animate Resources

Click the **Resource** button from the **Animate** toolbar. Select the resource of interest from the drop-down menu of the Identifier prompt. Create your own pictures or use the existing picture libraries. When finished, select **OK**. Your cursor will become a cross hair allowing you to click in the logic window where you would like the resource to be animated.



Animate the Simulation Time

Click the **Clock** button from the **Animate** toolbar. Define the attributes of the clock.



When finished, select **OK**. Your cursor will become a cross hair. To place a digital clock in the logic window, decide where you wish to place the upper-left corner of the box and click; then drag to the desired size and click to place the lower-right corner. To place an analog clock, first click the center of the circle followed by the outside diameter.



Resource Set Workshop (Resource Set Workshop.doe)

A Pittsburgh bank is interested in improving the customer service at one of their branch offices while also reducing operating costs. The bank has decided that simulation could help them analyze their system.

Bank management would like to model customer arrivals accurately. They also decided that they could provide the necessary customer service with two full-time tellers (each having a break at designated times), one part-time teller to help during peak hours, and one manager available for the entire day.

Currently, customers arrive at the bank and form a single line, waiting until a teller is available. Once a teller is available, the customer's transaction is first reviewed for the next plan of action. This process usually takes 1 minute, but can range between a low of 45 seconds and a high of 1.5 minutes.

Approximately 66% of the transactions require a simple transaction that the teller is able to perform alone. This simple transaction usually takes 1.5 minutes, but can take anywhere between 1 minute and 2.25 minutes to accomplish. After this transaction, the customer exits the bank.

The remaining transactions are more complex and require the assistance of the manager for approval. These transactions require between 2 and 3.5 minutes. After this type of transaction, as with the simpler transactions, customers exit the bank. (Hint: Use a "Delay Release" with a delay of zero to release resources.)

The tellers and manager are scheduled as follows:

- Two of the tellers should be on duty from 9:00 a.m. to 5:00 p.m.
- One of the tellers has a break from 11:00 a.m. to 11:30 a.m., while the other teller has a break from 12:45 p.m. to 1:15 p.m.
- The part-time teller will only work from 11:00 a.m. to 3:00 p.m. (Hint: Remember to add zero capacity at end of work day.)
- The manager is on duty from 9:00 a.m. to 5:00 p.m. as well and takes a break when time is available (i.e., don't worry about a schedule for the manager).
- Also, because the part-time worker is slightly less experienced, the manager would prefer that the part-time worker be the least-utilized teller.

Customer arrivals will follow the schedule below:

9:00 a.m. – 10:00 a.m.	20 customers per hour
10:00 a.m. – Noon	30 customers per hour
Noon – 2:00 p.m.	40 customers per hour
2:00 p.m. – 4:00 p.m.	30 customers per hour
4:00 p.m. – 5:00 p.m.	20 customers per hour

The base units should be in minutes for the 8-hour day.

Examine the utilizations of the resources and the time that the two different transactions take to flow through the system.

Run the simulation for one day.

ADDITIONAL EXERCISES:

- Add animation for the resources
- Place a clock for animation purposes
- Create a schedule for the manager with a break from 1:30 p.m. to 2:00 p.m.

5: Making Decisions with Simulation

Topics covered in this section:

Concepts

- Arena collected statistics, including average values, confidence intervals and half-widths
- Model time frames (Terminating and Non-Terminating)

Arena Constructs

- Arena reports (Category Overview)
- Run > Setup

Work the Model

A model is built to provoke thinking. A model can lead to system discovery. Insight comes from manipulating the model, not from the model itself!

Variable Input, Variable Output

Variable input induces variability in the output. You must interpret and cope with variability in the output.

For example, the results from a single "simulated day" are not sufficient because we really have no idea how "typical" the results are or how much variability might be associated with them. In statistical terms, this is referred to as a sample size of one, which isn't worth much!

Obtaining a confidence interval for any performance measure is a basic component of analyzing the model results.

Confidence intervals are generated from multiple independent and identically distributed data points.

Confidence Interval and Half Width

A confidence interval provides insight to the variability of the output. The larger the range, the more variation in the output.

It is expressed as a numerical range that has a probability of including the true value of the performance measure.

The half width is half the range of the confidence interval.

For appropriate statistics, Arena includes a category called "Half Width" in the reports. This category is included to help you determine the reliability of the results. This information can also be used to determine when to terminate a replication/run once a specified level of accuracy has been reached.

If a value is returned in the Half Width category, this value may be interpreted by saying "in 95% of repeated trials, the sample mean would be reported as within the interval sample mean ± half width."

Confidence Intervals

Given: An estimate for some system performance criteria, such as average time spent in the system.

The true (unknown) performance criteria will fall within a certain interval with a specified degree of confidence.

If the confidence level is fixed (e.g., 95%), a larger sample size will result in a smaller confidence interval.

If the size of the confidence interval is fixed (e.g., within 10% of the sample mean), a larger sample size will result in a higher degree of confidence.

If the sample size is fixed (e.g., 30 observations), a higher degree of confidence is associated with a larger confidence interval.

Model Time Frame

Most simulation models can be classified as either terminating or steady-state. This classification is based upon system characteristics and impacts the method used to generate a confidence interval.

Terminating

A terminating system is one in which there is a specific starting condition (typically empty) and a stopping condition.

Example: Servicing customers in a bank that is open between the hours of 9 a.m. and 5 p.m.

Steady-State

A steady-state system has no finite duration; the system is in perpetual operation.

Example: Processing insurance claims where claims remain in the "in-basket" if they are not processed by the end of a day.

NOTE: In most steady-state system analysis, a ***warm-up period*** is utilized. This period is the estimated point in time when a simulation run has reached steady-state. This is used to "clear" any statistics collected, such that only steady-state observations are used for statistical analysis.

Terminating System Analysis

The model run length is a given, and we are interested in how a performance measure is affected by the variation during that time period. Therefore, the system analysis is performed by running multiple replications.

The confidence interval of interest is generated using the average value (of the performance measure of interest) from each replication.

The category overview report provides basic statistics summarizing each data category (like cycle time) across all replications.

It provides:

Average — the simple (unweighted) average of the averages in each replication.

Half Width — the 95% confidence interval range about the average. Two results are possible in the "Half Width" category:

Insufficient: Arena will return the message "Insufficient" for that variable's half width if the number of replications is 2 or smaller.

A value: If a value is returned in the Half Width category, this value may be interpreted by saying "in 95% of repeated trials, the sample mean would be reported as within the interval sample mean ± half width."

Minimum Value — the lowest individual observation (not lowest average).

Maximum Value — the highest individual observation (not highest average).

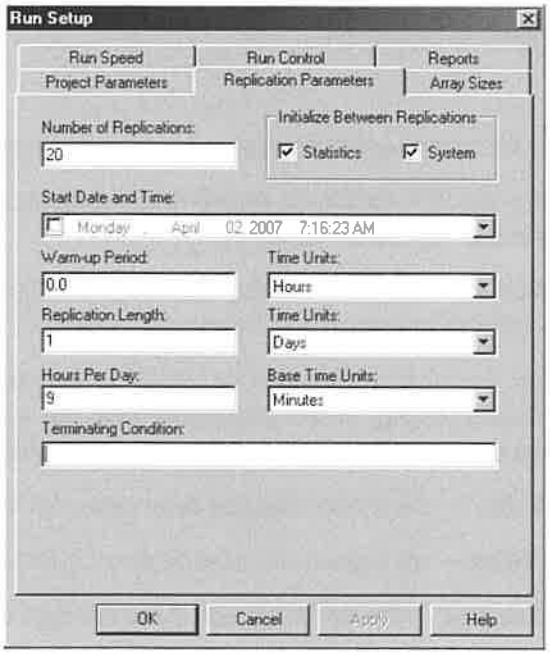
Minimum Average — the lowest average of the averages of each replication.

Maximum Average — the highest average of the averages of each replication.

Terminating System Analysis — (DMV Photo License Center – Part 2.doe)

The Photo License Center is a terminating system since the center opens up empty every morning (without customers) and idle (resources not busy) and then closes empty as well.

To obtain the half width of the confidence interval on a system parameter, we need to make multiple replications of a simulated day.



For example, let's look at the cycle time through the DMV facility for the drivers.

Total Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
ent Drivers	11.1083	1.46	6.2927	16.9005	3.4787	41.8116

NOTE: Check the Batch Run option on the **Run > Run Control** menu to turn off all animation for the fastest simulation run.

Steady-State System Analysis

There are no defined starting and ending points for the system. We are interested in the steady-state behavior. Therefore, we run the model for a long period of time. We may also consider clearing statistics after a given warm-up period.

The confidence interval for our performance measure of interest is generated from "batch averages" calculated from a single long run.

The detail reports on entities, processes, queues, and resources provide basic statistics summarizing each data category (like cycle time) for every replication.

It provides:

Average — the simple (unweighted) average of the replication.

Half-width — the 95% confidence interval range about the average. Three results are possible in the "Half Width" category:

Insufficient: Arena will return the message "Insufficient" for that variable's half width, indicating there is insufficient data to calculate the half width accurately. Running the replication for a longer period of time should correct this.

Correlated: The formula used to calculate half width requires the samples to be independently distributed. Data that is correlated (the value of one observation strongly influences the value of the next observation) results in an invalid confidence interval calculation. If data is determined to be correlated, the message "Correlated" is returned for that variable's half width. Running the simulation for a longer period of time should correct this.

A value: If a value is returned in the Half Width category, this value may be interpreted by saying "in 95% of repeated trials, the sample mean would be reported as within the interval sample mean ± half width."

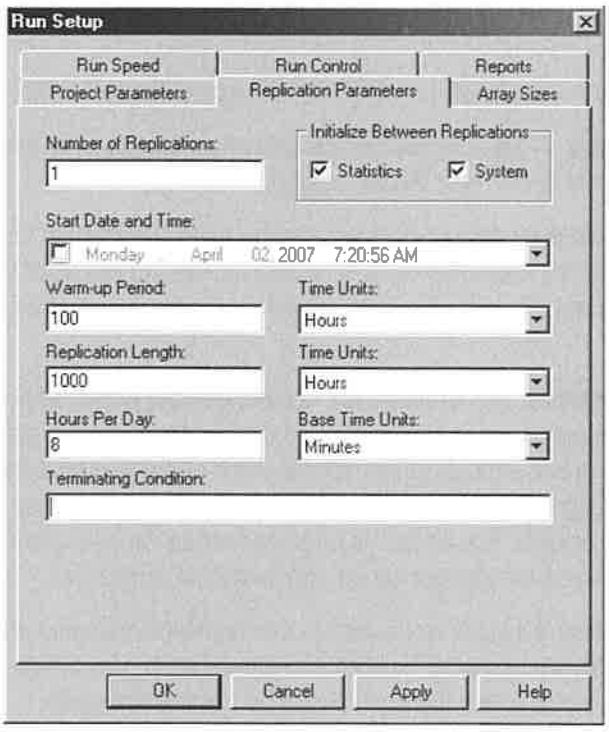
Minimum — the lowest individual observation for the replication.

Maximum — the highest individual observation for the replication.

Steady-State System Analysis – Auto Loan Part 2.doe

To analyze the loan-processing model, we need to treat the process as a steady-state system. The system does not start empty every morning. When the agents and clerks go home at the end of the day, they may leave in the middle of processing an application. However, upon arrival the next day, they will start up where they left off.

To obtain the half width of the confidence interval on a system parameter, we need to make only one long run.



For example, let's look at the total time for the loan applications.

Total Time	Average	Half Width	Minimum Value	Maximum Value
ent Auto Loan Applications	90.2213	3.90277	12.2834	197.04

Decisions with Simulation Workshop (Decisions with Simulation Workshop.doe)

Recently, the Purchasing Department from the Basic Process workshop has seen a dramatic increase in the amount of purchase requests. Requests now arrive approximately every 5 minutes. The company has hired four additional Purchasing Agents to try to satisfy the additional requests.

Make the system changes to the simulation created in the Basic Process workshop, and save the model under a different name.

Determine the type of system for this model (i.e., terminating or steady-state). Based on the system type, perform run analysis such that the half-width of the Purchasing Agents utilization is within +/- 5% of the mean. For this workshop, do not be concerned with a warm-up period.

6: OptQuest® for Arena®

Topics covered in this section:

Concepts

- Creating optimization model

Arena Constructs

- Tools > OptQuest

OptQuest enhances the analysis capabilities of Arena by allowing you to search for optimal solutions within your simulation models. OptQuest automates, or controls, Arena to set variable values, start and continue simulation runs, and retrieve simulation results.

OptQuest is a generic optimizer that makes it possible to separate successfully the optimization solution procedure from the simulation model. The optimization procedure uses the outputs from the simulation model to evaluate the inputs to the model. The optimization procedure then uses a search algorithm, where the successively generated inputs produce varying evaluations, not all of them improving, but which over time provide a highly efficient path to the best solutions. The process continues until it reaches some termination criterion.

OptQuest uses the elements of controls, responses, constraints, and objectives in searching for optimal solutions. These elements are discussed in the following section.

Concepts and Terminology

Controls

Controls are system elements that can be meaningfully manipulated to affect the performance of the system (e.g., number of workers, starting inventories, reorder points, etc.). In terms of OptQuest, controls are Arena Variables or Resources defined within the model. OptQuest varies the values of these controls (based on a range specified) to meet a desired outcome (e.g., maximize throughput, minimize cost). The values of the controls are changed before each simulation is performed until the best values are found within the allotted time limit.

Responses

Responses are outputs of the simulation model, including: tallies, outputs, CSTATS, DSTATS, counters, and variables. In many cases, the objective function and constraints depend on the response values.

Constraints

Constraints are relationships among controls and/or responses. Constraints put limits on whether control and/or response values enable the model to produce feasible results. For example, a constraint might ensure that the total number of workers does not surpass a certain sum, or ensure that the maximum wait time within a queue does not surpass a certain value. Specifying constraints is not necessary in optimization modeling.

Objective

An objective is a mathematical response or an expression used to represent the desired goal of the simulation (in terms of the assumptions and controls). Each optimization model has one objective function that is used to either to minimize or maximize the mathematical response. Some examples of objectives include: minimizing queues, maximizing profits, minimizing cycle time, maximizing throughput, etc.

OptQuest searches for the optimal value of the objective by selecting and improving different values for the controls.

Model Design Considerations

- Controls must be either Resources or Variables within the simulation.

TIP: If using resource schedules with a resource as a control, create the resource schedule by using variables (for the duration and capacity values) and have the variable values be controls.

- Since OptQuest changes control values prior to the start of each replication, do not change the values during the model run itself. For example, do not read in variable values that you wish to use as controls. If you do this, the reading action will overwrite the values that OptQuest has modified.

TIP: Modify variable values that are used as controls in Arena Edit mode. This can be performed manually or through use of VBA.

- Eliminate or disable anything within the model that requires user interaction (e.g., Message boxes).

OptQuest Exercise (OptQuest Training Example.opt)

The manager of the purchasing department (Decisions with Simulation Workshop) would like to control the use of her resources while optimizing the number of complete requests. The requirements of the system are as follows:

The manager has control of the number of Interns and Purchasing Agents she can employ. She has determined that she has to have at least one but no more than three interns. Also, based on union rules, she must employ at least six but no more than 11 Purchasing Agents. Partial use of employees is not permitted.

Due to budgetary constraints (not reflected in the model), she can have a maximum of 12 employees working for her. Also, to reduce customer dissatisfaction, the average wait time for the agent to fill the order must not exceed two minutes.

The overall goal of the optimization is to maximize the number of completed requests.

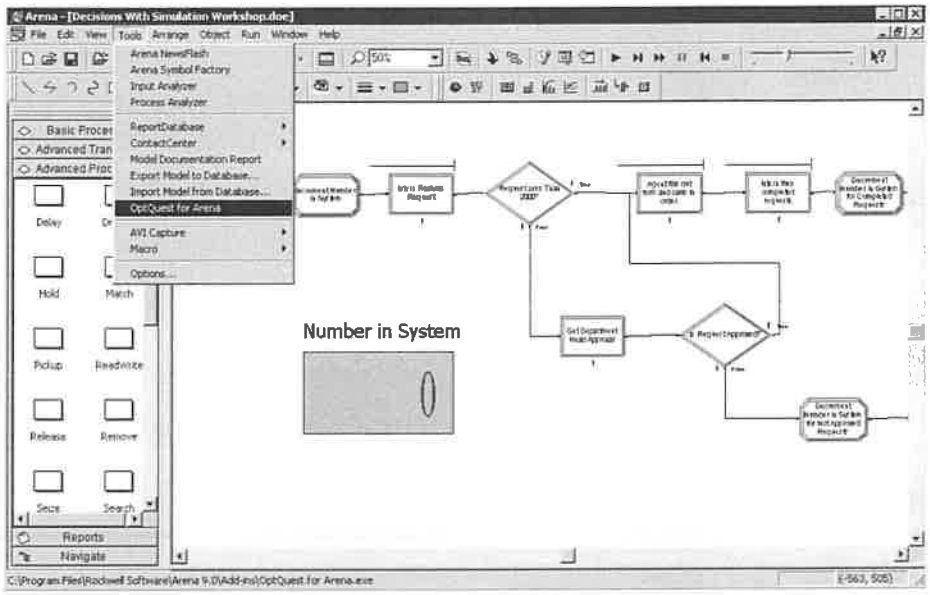
Use the Decisions with Simulation Workshop (*Decisions with Simulation Workshop.doe*) and OptQuest to determine the optimal number of Interns and Purchasing Agents based on the goal of the optimization.

Hint: Create two responses:

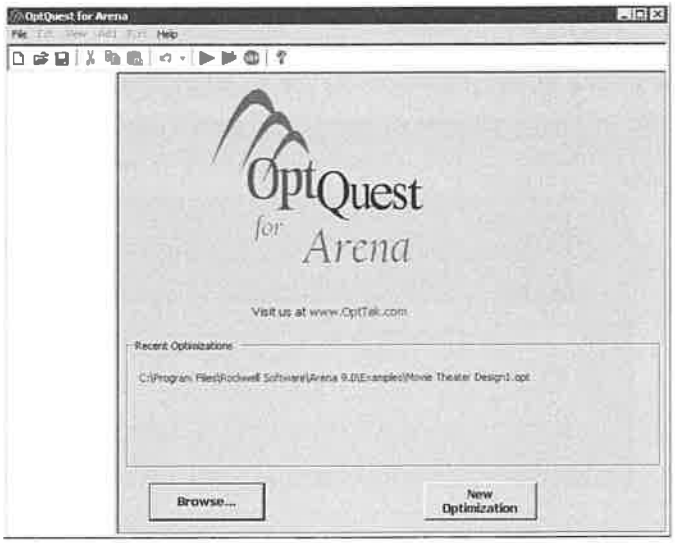
1. Entity > Completed Requests.NumberOut
2. Queue > Waiting Time > Agent Fills out form and calls in order.queue.WaitingTime

Steps for Using OptQuest:

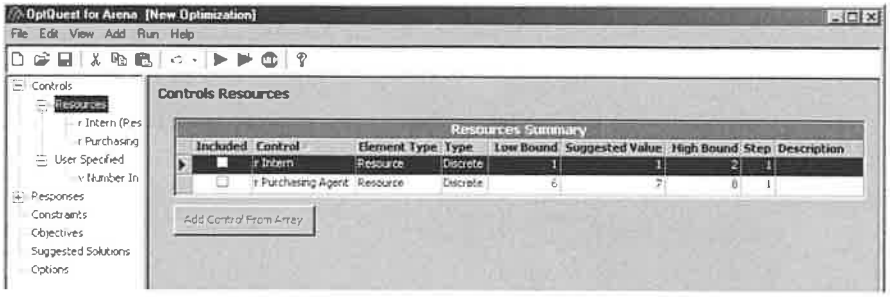
1. Select OptQuest from the main menu via **Tools > OptQuest.**

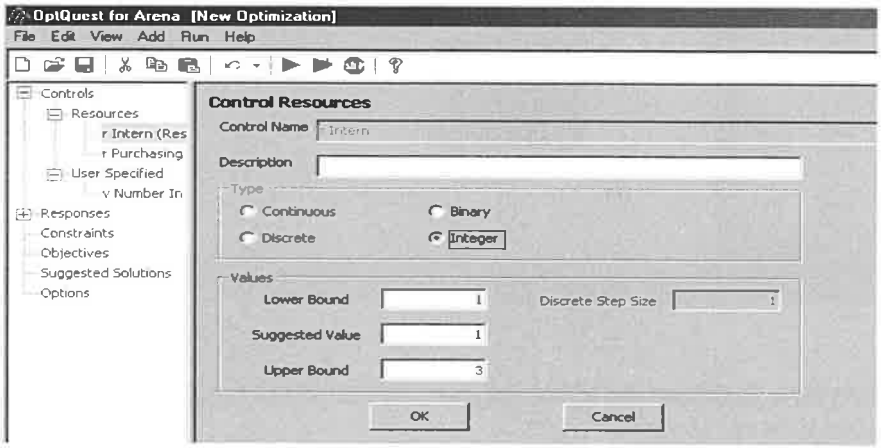


2. Select **New Optimization** within OptQuest. This will check the current model and place the model in Run Mode.

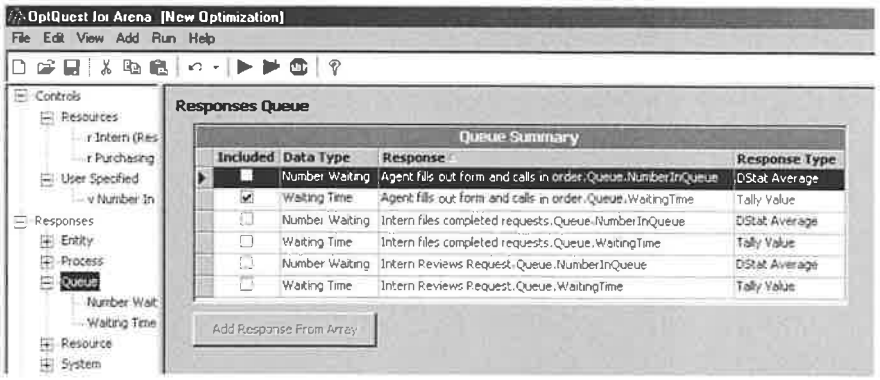


3. Choose Controls branch. All of the Variables and Resource from the model are now displayed. Select the Intern and Purchasing Agent, setting the Lower and Upper Bounds for each.

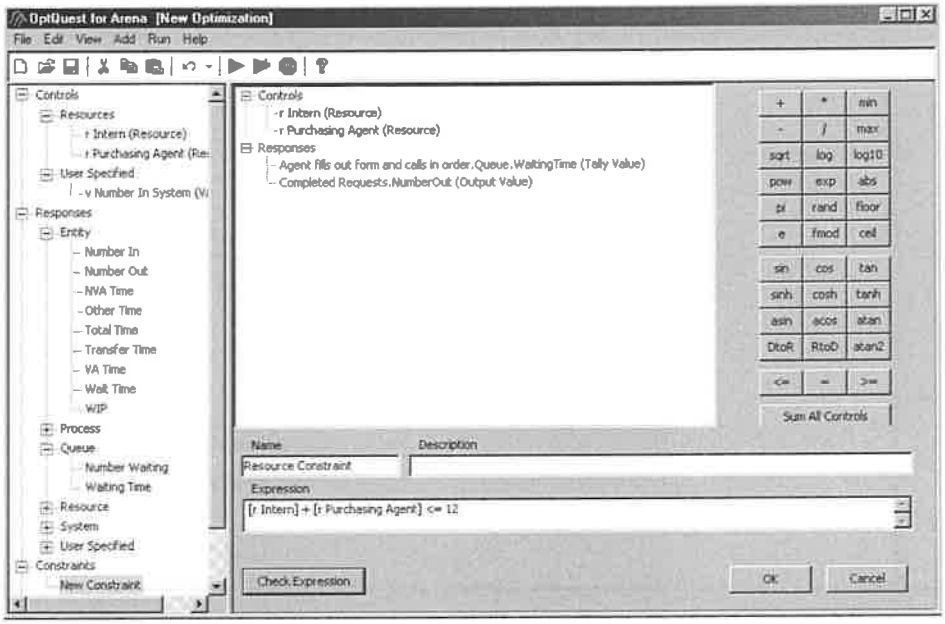


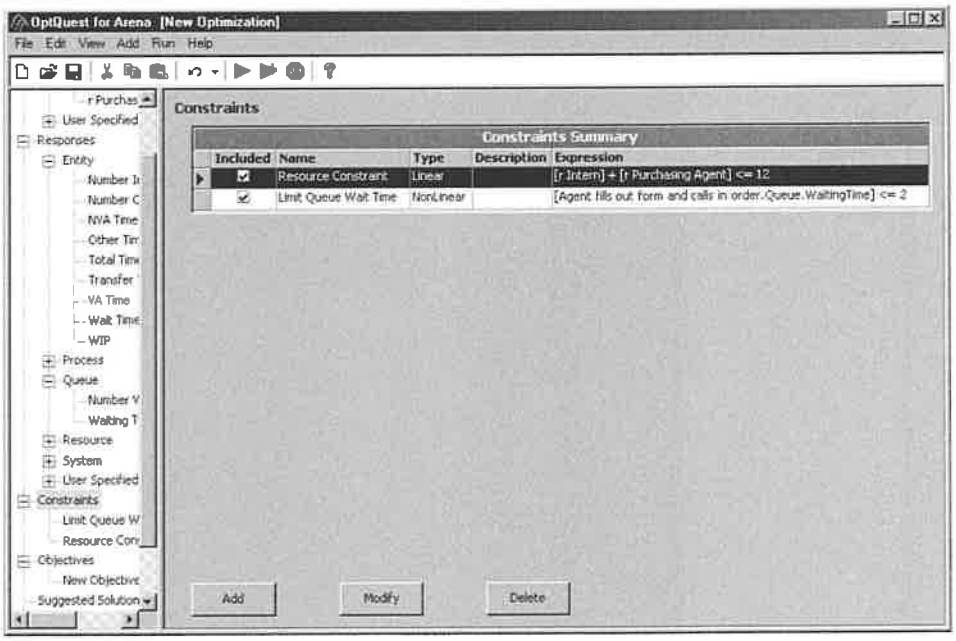


4. Expand and select the appropriate Responses from the Responses branch. These responses will be used in setting the constraints and for the objective function.

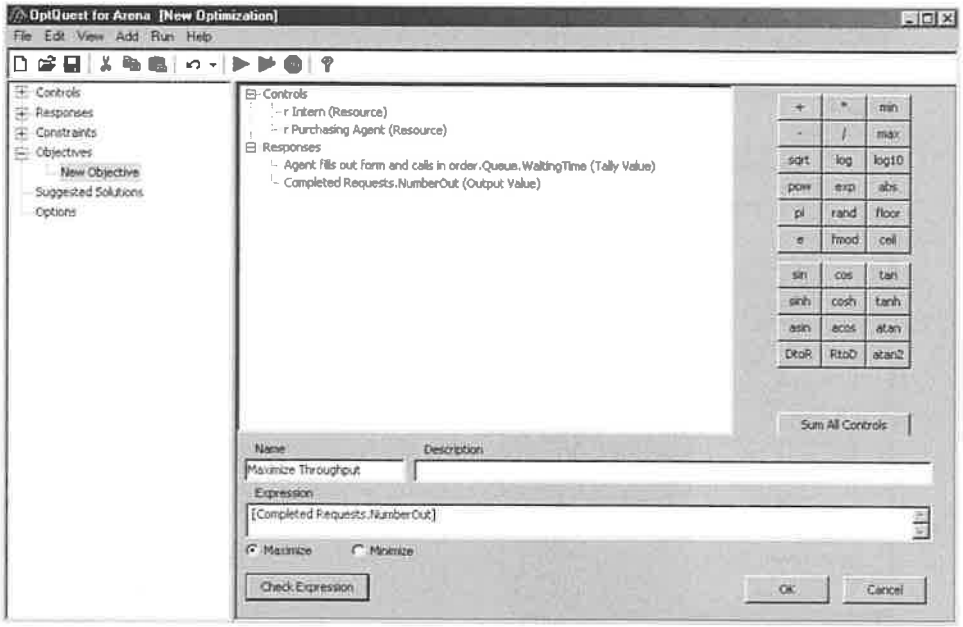


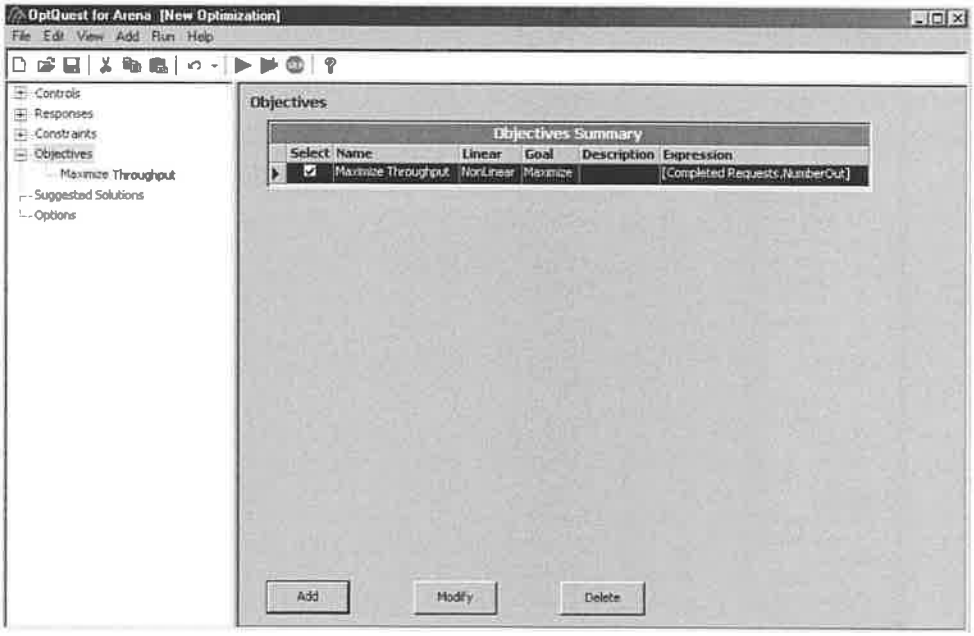
5. Select the Constraints branch. Use the **Add** button to create the constraints to limit the total number of resources and the maximum queue time.



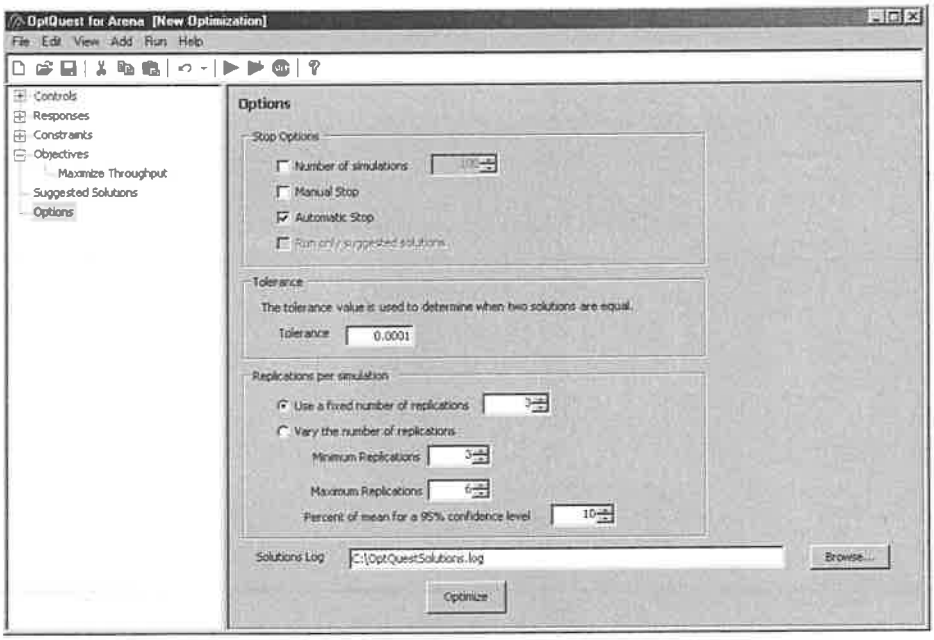


6. Select Objective branch and create the objective function to maximize the throughput of requests.

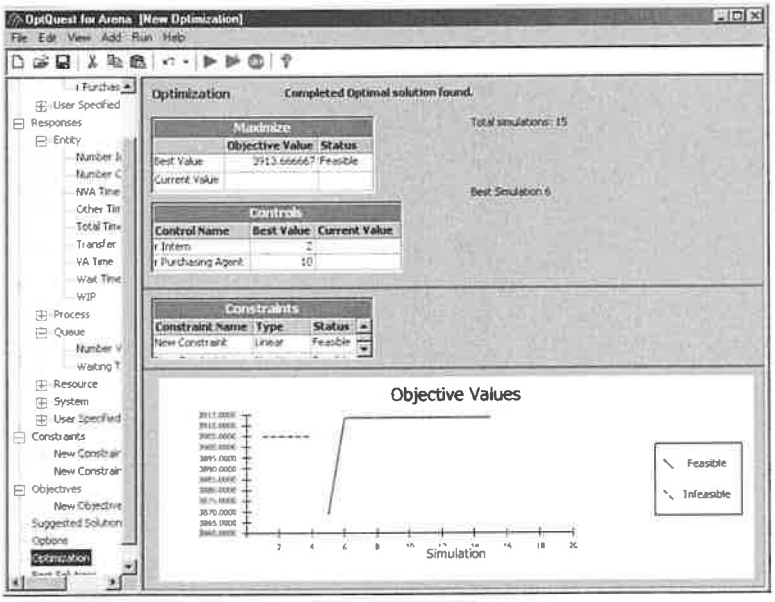




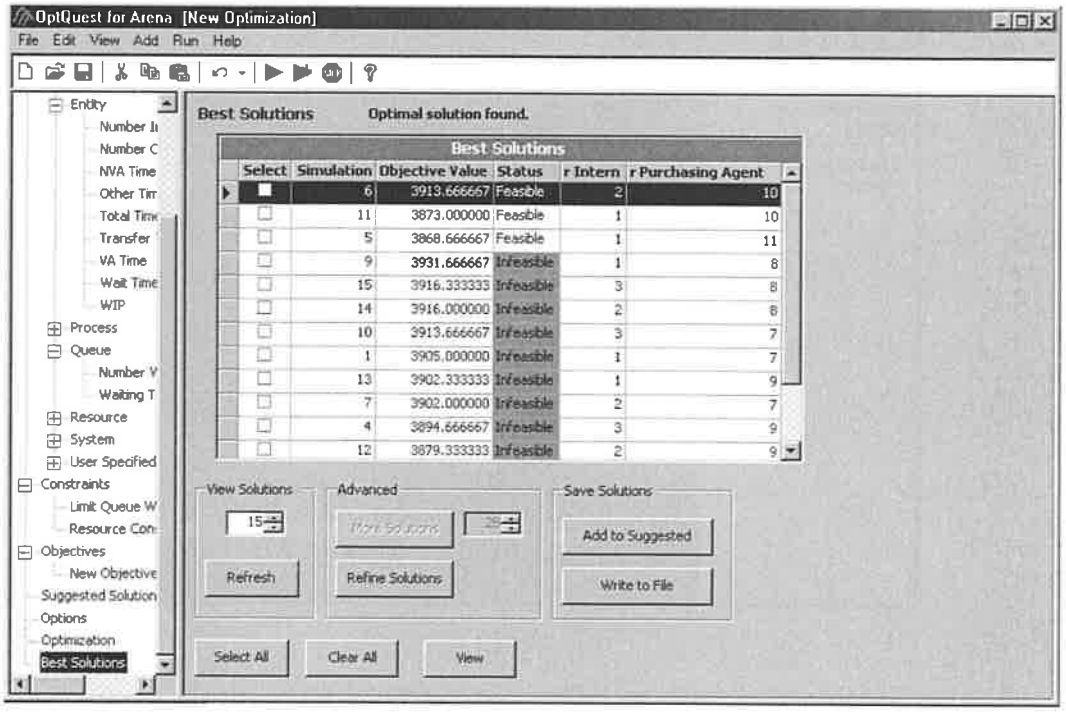
7. Select Options branch to configure optimization run parameters. Run for 3 replications. Press **Optimize** to start the optimization.



8. View the optimization results as OptQuest searches for the optimal result.



9. After the optimization has been complete, view the results in Best Solutions window. Notice the Infeasible results.



7: Process Modeling

Topics covered in this section:

Concepts

- Queue ranking
- Unexpected resource downtimes
- Creating user-specified statistics

Arena Constructs

- Seize logic module
- Delay logic module
- Release logic module
- Queue spreadsheet module
- Failures spreadsheet module
- Resource spreadsheet module
- Statistic spreadsheet module

Since systems can vary greatly in how they operate, Arena provides tools that can help model your complex system actions.

If the module constructs provided in the Basic Process panel do not capture the system logic at the required level of detail, the module constructs in the Advanced Process and Advanced Transfer panels provide the ability to model complex systems actions. These panels add enormous power and flexibility to your ability to represent a system.

The exercises discussed in this section illustrate the functionality of commonly used modules contained in the Advanced Process and the Advanced Transfer panels.

Concepts and Terminology

Seize, Delay, and Release

Up to this point, all of our resources have been seized, delayed, and released using the options contained within the Process module. In order to facilitate more complex models, the Advanced Process panel contains three separate modules for each of the actions.

The Seize module allows an entity to take control of a resource. This module can be used to halt the entity's movement through the system until a desired resource is available.

The Delay module causes the entity's movement through the system to be delayed for a specified amount of time. When used without a Seize module, the Delay module will allow an infinite number of entities to enter and be delayed.

The Release module allows an entity to relinquish control of a resource. This module is always used in conjunction with a Seize module. If an entity tries to release a resource that is not busy, Arena will generate an error.

For all of the following examples, the Seize, Delay, and Release modules will be used in place of the Process module. However, feel free to use either in your models.

Queue Ranking

Sometimes work is not completed in a First In, First Out manner. To allow you to model this situation, Arena provides the ability to rank the entities waiting in a queue based on several rules.

First In, First Out — Entities entering a queue that try to seize a resource will be ranked according to the time that they entered the queue. The first entity to enter the queue will be the first entity in line to seize the resource. Entities will not "jump" line. This is Arena's default.

Last In, First Out — Entities entering a queue that try to seize a resource will be ranked according to the time that they entered the queue. The most recent entity to enter the queue will be the first entity in queue.

Lowest Attribute Value First — Entities entering a queue that try to seize a resource will be ranked according to a user-specified attribute with the lowest value given priority.

Highest Attribute Value First — Entities entering a queue that try to seize a resource will be ranked according to a user-specified attribute with the highest value given priority.

Failures

By default in Arena, a resource is available 100% of the simulation time. In reality, resources usually will be unavailable due to a variety of reasons such as a tool breaking on a machine, a worker leaving the workstation for a few minutes, a power outage, or any other reason.

These unexpected losses of a resource can be modeled using failures.

A failure can remove a resource based on a time or count.

The Preempt, Wait, and Ignore options for failures of a busy resource work similarly to these options in the schedule spreadsheet.

Reflected in failed statistic (Frequencies section).

You may define multiple failures for a resource.

Statistics

The Statistic data module is used to define additional statistics that are to be collected during the simulation and also to specify output data files.

In Arena, many statistics are automatically collected directly within a module. However, you may want to collect additional statistics, such as frequencies for the state of a resource or more complicated time-persistent statistics.

The types of statistics that may be defined in the Statistic module are time-persistent, tallies (observational data), count-based, outputs, and frequency-based.

Basic statistics are also available to be saved to an output file. Statistics saved to output files have each individual observation (e.g., each tally recorded) written to the output file. This output can later be used for data analysis.

These statistics are displayed in the Frequencies portion of the Reports section.

Process Modeling Exercise 1 (Process Modeling Exercise 7-1.doe)

The manager responsible for the Toy Paint Shop of the Toy Manufacturing and Distribution Company (TMDC) has approached you to build a model of his current paint process. Management at TMDC is looking into adding additional toy resellers. Before management finalizes the contracts with the new resellers, output from the TMDC toy production facilities must meet the additional demand for toys. The Toy Paint Shop manager has been tasked to increase output and would like to use the model to look for bottlenecks and test changes to the system.

The Toy Paint Shop processes two different toy part types. The part types enter a paint shop from an inventory room where they undergo several different processes before exiting the system. The time between arrivals for the first Part Type, a toy boat, is exponential with a mean of 5 minutes. The time between arrivals for the second Part Type, a toy truck, is exponential with a mean of 8 minutes.

Both part types then enter a Paint Preparation process where they must seize one of two operators in order to complete the preparation process. Once the Prep operator has been seized, the parts require anywhere from 2 to 8 minutes to process. Priority is given to toy boats.

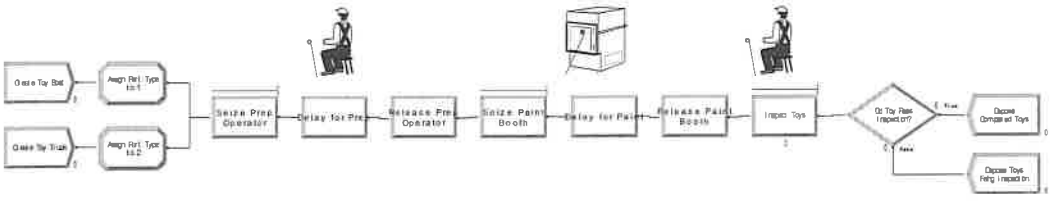
After the preparation step, the parts must be painted in a Paint Booth. This process takes exactly 2.5 minutes and can process one part at a time. Parts are processed first come, first served. The Paint Booth must be cleaned after 100 parts are painted. This cleaning process can take anywhere from 5 to 10 minutes, but usually requires 7 minutes.

Once a part has been painted, a single inspector determines if the part was manufactured and painted correctly. This inspection can take anywhere from 2 to 4 minutes, but usually requires 3 minutes. It is estimated that 90% of the toy parts pass the inspection.

Once a part is inspected, it will exit the system.

Run the model for five 8-hour days. The base time units should be minutes.

Examine part cycle-time statistics, number in queue statistics, resource utilization statistics, and frequency statistics.



Modeling Approach

Part Arrival: Use two Create modules to generate arriving toys and use Assign modules to set attribute Part Type.

Prep: Use the Seize, Delay, and Release modules to model the Prep Operators' process.

Painting: Use the Seize, Delay, and Release modules to model the Paint Booth.

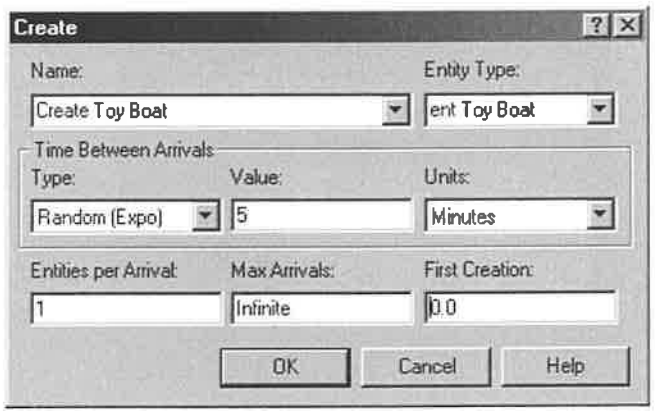
Paint Booth Cleaning: Add a count-based failure to the Paint Booth resource. Define the downtime using the Failures option in the Advanced Process panel. Capture the frequency that the resource is in its states by using the Statistic module in the Advanced Process panel.

Inspection: Use a Process module and a Decide module to model the Inspection process.

Complete Parts: Use two Dispose modules to model complete parts and the parts failing inspection exiting the system.

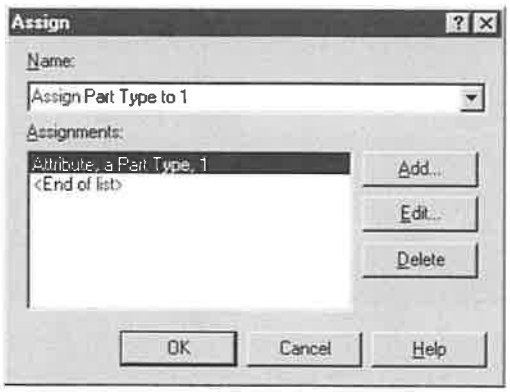
Run Length, Part Animation: Use the **Run > Setup** option to specify the run length and time units. Use the animation tools to model the resources.

Toy Boat Arrival — Use the Create module



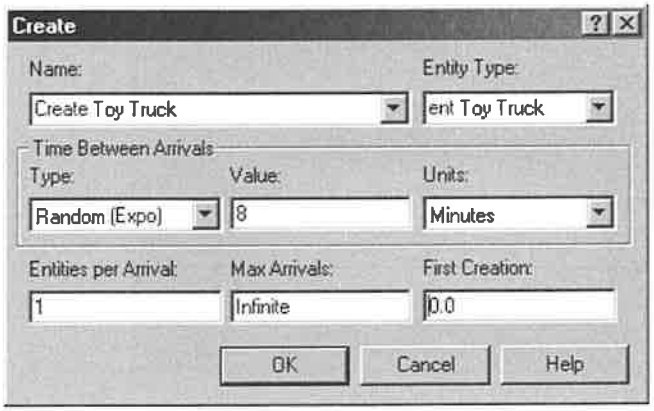
Name	Create Toy Boat
Entity Type	ent Toy Boat
Time Between Arrivals	Random (Expo) – 5 minutes
Assign initial picture in Entity Spreadsheet	Picture.Boat

Assign Part Type to 1



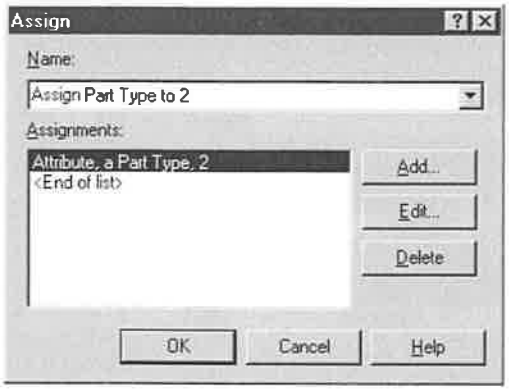
Name	Assign Part Type to 1
Attribute Name	a Part Type
New Value	1

Toy Truck Arrival — Use the Create module



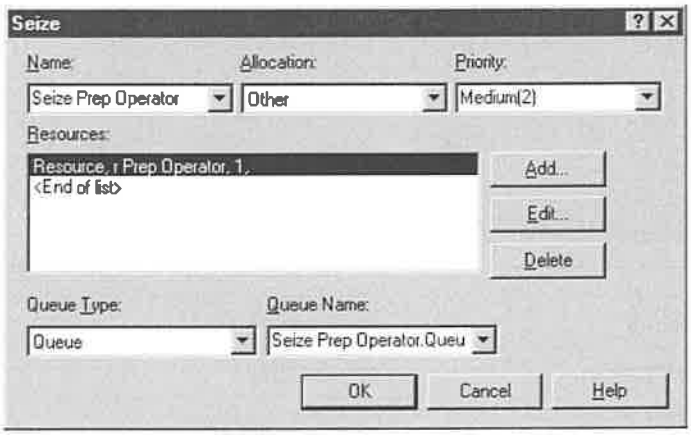
Name	Create Toy Truck
Entity Type	ent Toy Truck
Time Between Arrivals	Random (Expo) – 8 minutes
Assign initial picture in Entity Spreadsheet	Picture.Truck

Assign Part Type to 2



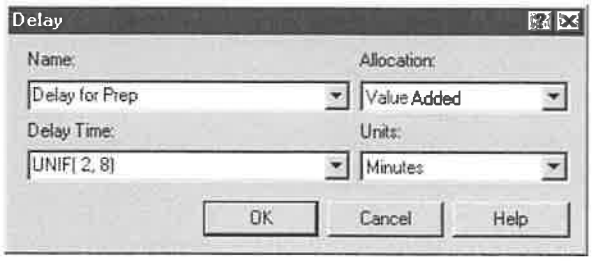
Name	Assign Part Type to 2
Attribute Name	a Part Type
New Value	2

Seize the Prep Operator — Use the Seize module



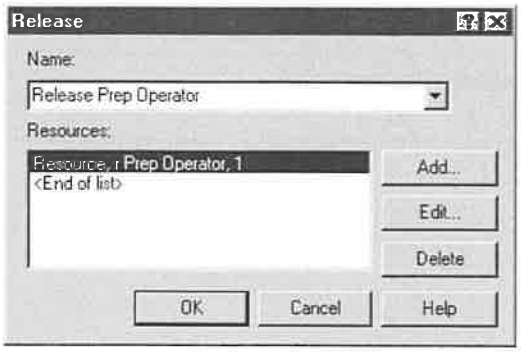
Name	Seize Prep Operator
Resources	r Prep Operator
Queue Type	Queue
Assign Capacity in Resource Spreadsheet	2

Delay for Preparation — Use the Delay module



Name	Delay for Prep
Allocation	Value Added
Delay Time	UNIF(2,8) Minutes

Release the Prep Operator — Use the Release module

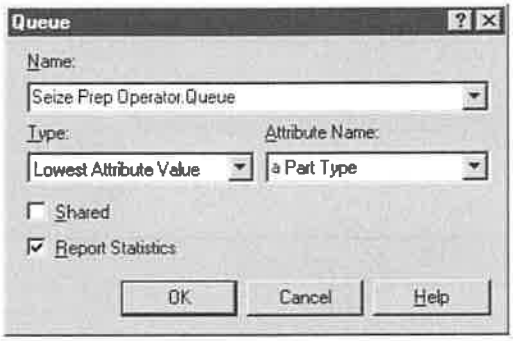


Name	Release Prep Operator
Resources	r Prep Operator

Define the Priority Queue for the Prep Process

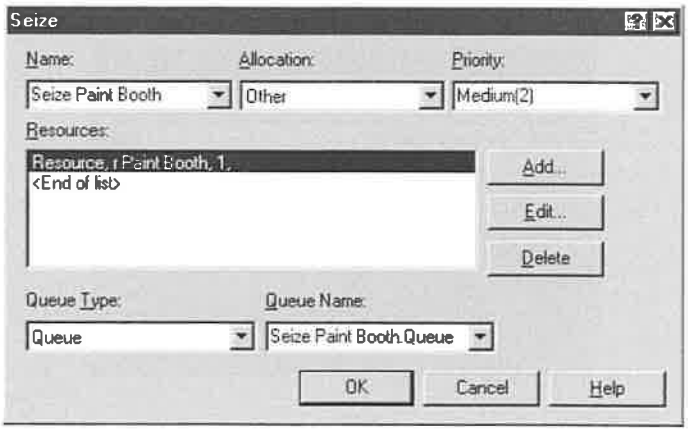


Queue



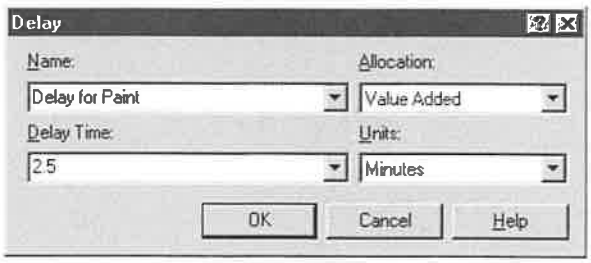
Name	Seize Prep Operator.Queue
Type	Lowest Attribute Value
Attribute Name	a Part Type

Seize the Paint Booth — Use the Seize module



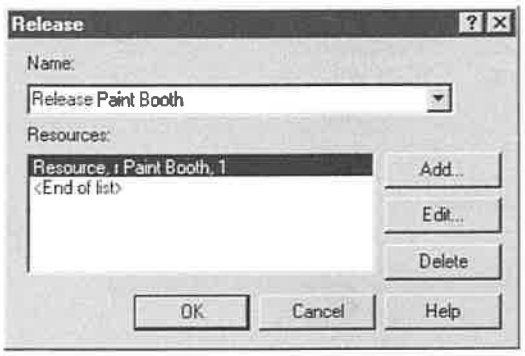
Name	Seize Paint Booth
Resources	r Paint Booth
Queue Type	Queue

Delay for Painting — Use the Delay module



Name	Delay for Paint
Allocation	Value Added
Delay Time	2.5 Minutes

Release the Paint Booth — Use the Release module



Name	Release Paint Booth
Resources	r Paint Booth

Define the Paint Booth Cleaning Operation



Resource

Resource - Basic Process

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	r Prep Operator	Fixed Capacity	2	0.0	0.0	0.0		0 rows	☑
2	r Paint Booth	Fixed Capacity	1	0.0	0.0	0.0		1 rows	☑
3	r Inspector	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑

Double-click here to add a new row

Paint Booth Failures – 1 row

Failures

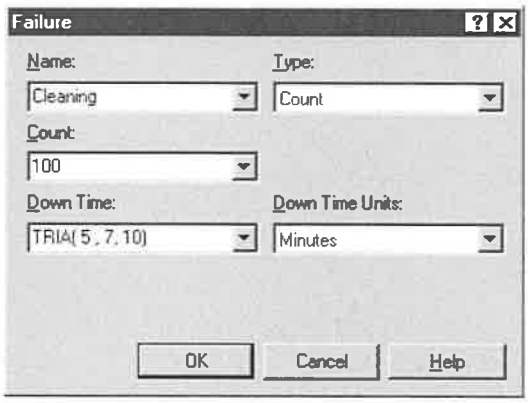
	Failure Name	Failure Rule
1	Cleaning	Ignore

Double-click here to add a new row

From Advanced Process panel



Failure



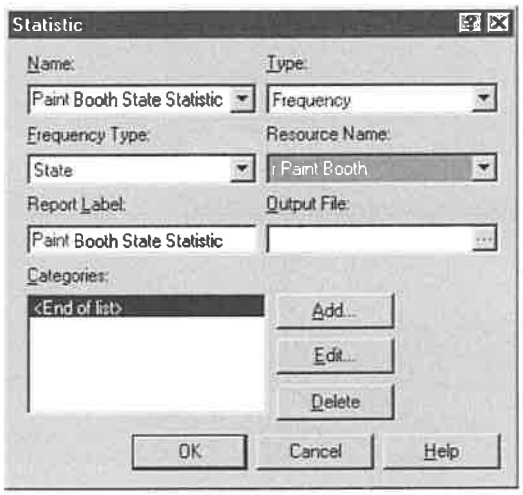
Name	Cleaning
Type	Count
Count	100
Down Time	TRIA(5,7,10) Minutes

Collect the Frequency Statistic for the Cleaning Operation

From Advanced Process panel

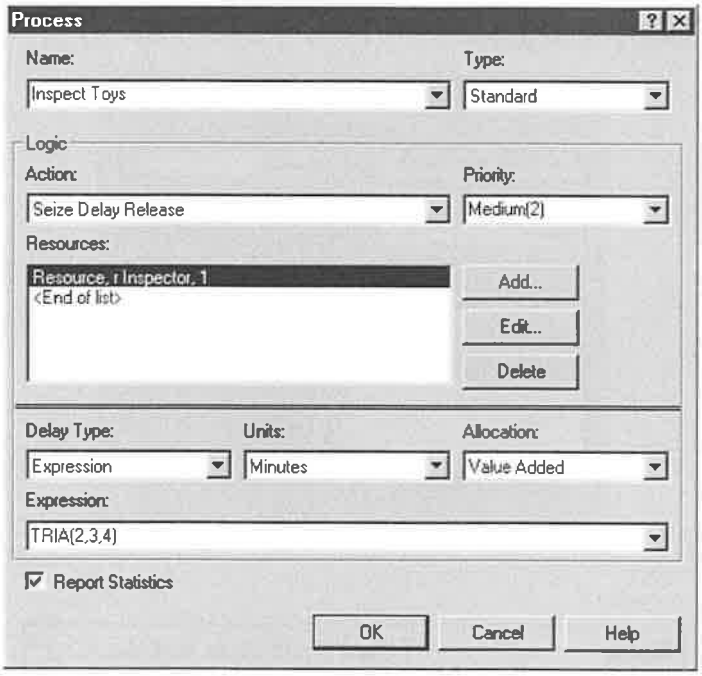


Statistic



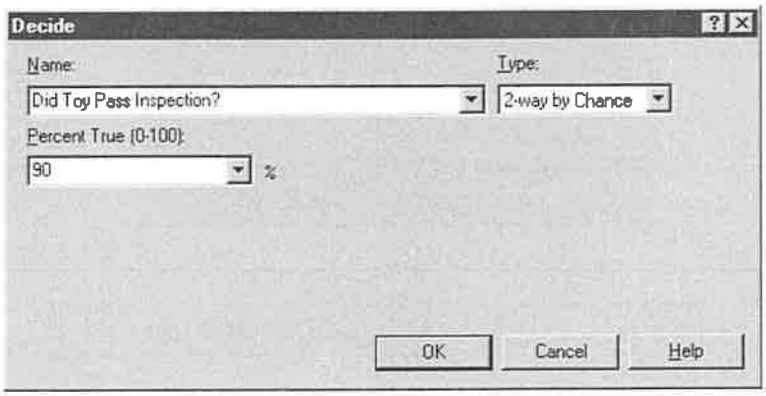
Name	Paint Booth State Statistic
Type	Frequency
Frequency Type	State
Resource Name	r Paint Booth
Report Label	Paint Booth State Statistic

Inspect Toys — Use the Process module



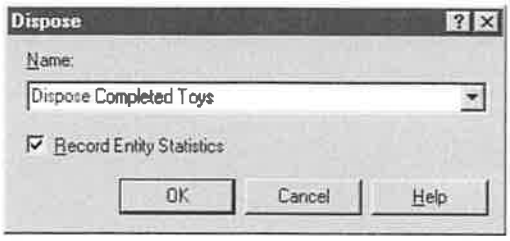
Name	Inspect Toys
Action	Seize Delay Release
Resources	r Inspector
Delay Type	Expression
Units	Minutes
Allocation	Value Added
Expression	TRIA(2,3,4)

Did Toy Pass Inspection? — Use the Decide module



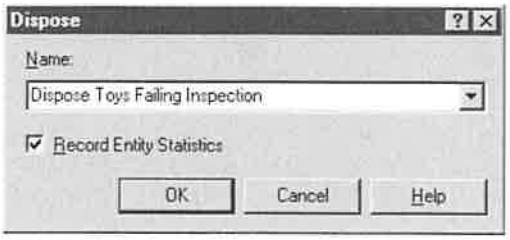
Name	Did Toy Pass Inspection?
Type	2-way by Chance
Percent True	90

Dispose Completed Toys — Use the Dispose module



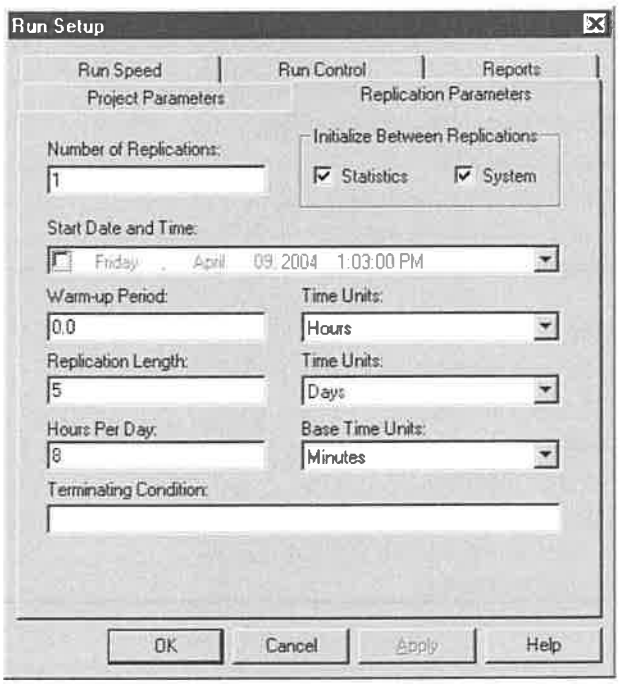
Name	Dispose Completed Toys

Dispose Toys Failing Inspection — Use the Dispose module



Name	Dispose Toys Failing Inspection

Run Setup



Number of Replications	1
Replication Length	5 Days
Hours Per Day	8
Base Time Units	Minutes

Topics covered in this section:

Concepts

- Unconstrained movement of entities through system
- Duplicating and ungrouping entities
- Grouping of entities
- Creating own entity pictures
- Animation of entity movement
- Navigating through logic flow

Arena Constructs

- Station logic module
- Route logic module
- Separate logic module
- Batch logic module
- Assign logic module
- Animate Transfer toolbar (Station, Route)
- Edit > Entity Pictures
- View > Named Views

Concepts and Terminology

When building a simulation model, it is often critical to model the movement of entities through a system. This movement could be a simple delay or may require material-handling resources like conveyors or transporters.

Arena has a set of tools to help you model entity movement and material-handling devices. The Advanced Transfer panel contains modules to help you capture the details of your material-handling systems. The next exercise introduces the simple delay (unconstrained) routing functionality in Arena (Routes and Stations).

Stations

Arena approaches the modeling of entity movement by dividing the system into subsystem locations or stations.

Stations are used to represent processing areas of the system being modeled.

Stations...

- Make the modeling effort more manageable
- Provide a framework for the control of entity flow
- Provide a means to model entity movement

For example: a machining area, a service counter, a warehouse, etc.

Routes

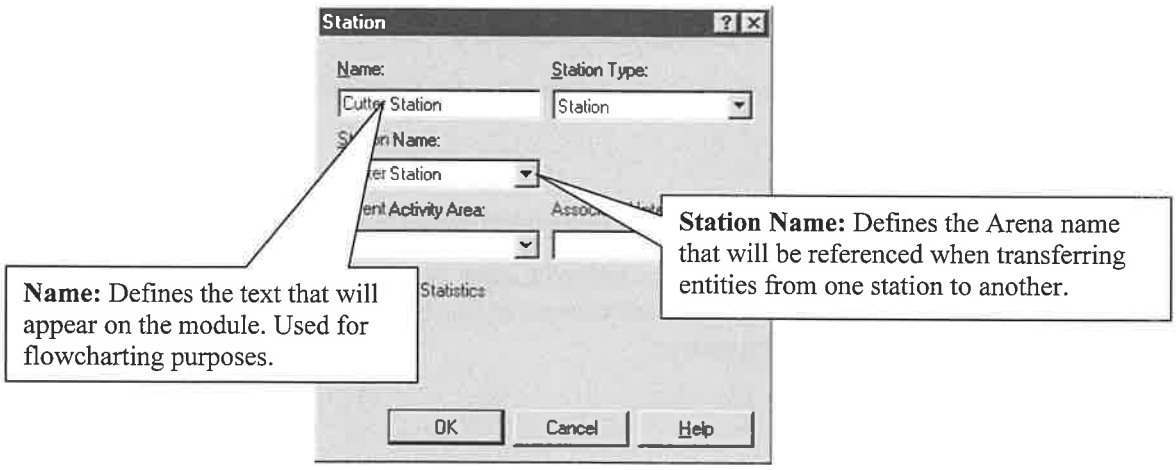
The unconstrained movement of entities from one station to another.

Routing assumes that time may be required to move the entity between stations, but it assumes that no additional delay will be incurred because of unavailable constraints, such as material-handling devices.

Special-Purpose Station Attribute

Arena provides an attribute, Entity.Station (M), to store an entity's current station location (or destination when in transit).

Module Notes: The "Name" field is the name of the module, used for flowcharting purposes only. The "Station Name" field is the name referenced when sending entities from one station to another. Also note, although not required, the "Name" and the "Station Name" may be identical. This is one of the few instances in Arena where two items may have the same name.



Separate Module



This module is used either to copy an incoming entity into multiple entities or to split a previously batched entity. Rules for allocating costs and times to the duplicate or split entities are also specified.

Batch Module



This module is a grouping mechanism. Batches can be permanently or temporarily grouped. Temporary batches must later be split using the Separate module.

Assign Module — Revisited



Previously this module was used for modifying a user-defined variable. This module can also be used for assigning new values to other items, such as entity attributes, entity types, entity pictures, or other system variables.

Process Modeling Exercise 2 (Process Modeling Exercise 7-2.doe)

Based on the results of the first simulation of the Toy Paint Shop, the manager wants to examine what would happen if parts from the painting operation were batched into like groups of two before they are inspected.

Parts may only be batched with another part of the same type. The two parts can then be inspected together in a time between 4 to 6 minutes.

NOTE: The decision for passing/failing a part should still be independent of the batch.

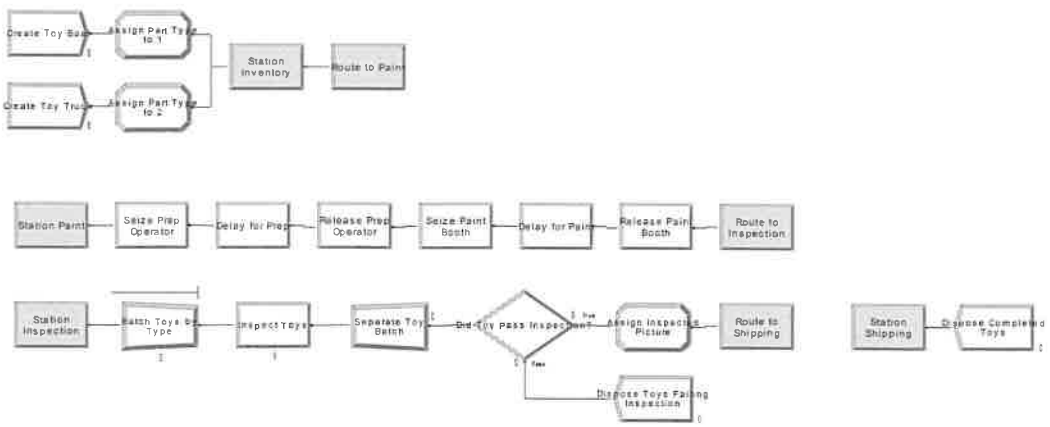
The manager has also asked that the model include the delays for the part handoffs between process areas as well as the animation reflecting the part handoffs. In addition, it is requested that *a unique entity picture be created for toys that have successfully completed inspection.*

Estimated routing delays provided by the manager:

From Station	*To Station*	*Move Time (minutes)*
Inventory Station	Paint Station	Constant (2)
Paint Station	Inspection Station	Constant (3)
Inspection Station	Shipping Station	Constant (3)

Run the model for five 8-hour days. The base time units should be minutes.

Examine part cycle-time statistics, number in queue statistics, and resource utilization statistics.



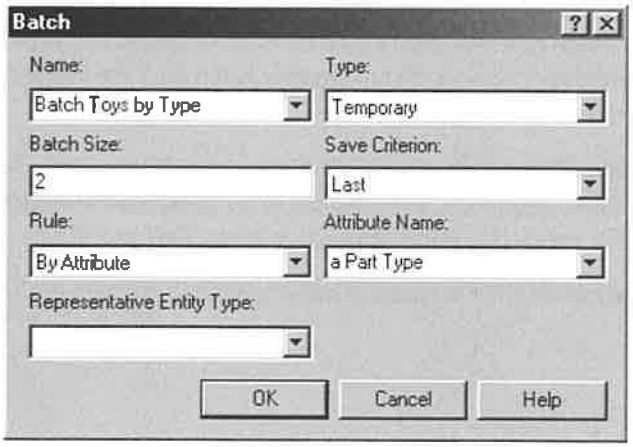
Modeling Approach

Batching Two Toys: Use the Batch module to batch together two similar parts based on entity type or the part type attribute. Separate batch after the inspection process using the Separate module.

Routing between Process Areas: Create Stations for the Inventory, Paint (Preparation and Paint processes), Inspection, and Shipping areas using Station modules from the Advanced Transfer panel. Use Route modules to model the handoffs.

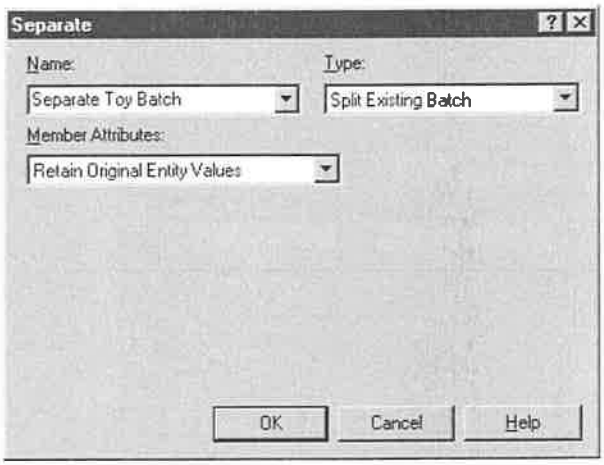
Routing Animation: Use the Animate Transfer toolbar to animate the Toy Paint Shop.

Batch Two Similar Parts — Use the Batch module



Name	Batch Toys by Type
Type	Temporary
Batch Size	2
Save Criterion	Last
Rule	By Attribute
Attribute Name	Entity.Type or a Part Type

Separate Parts — Use the Separate module

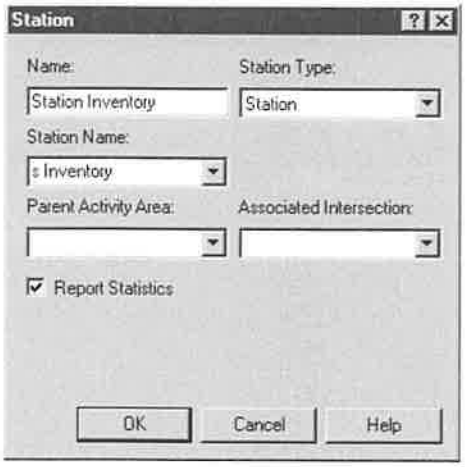


Name	Separate Toy Batch
Type	Split Existing Batch
Member Attributes	Retain Original Entity Values

Inspect Toys — Update the Process module

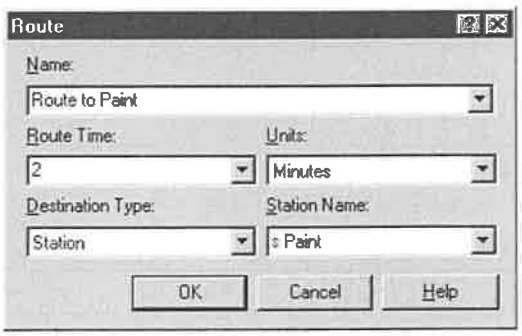
Name	Inspect Toys
Expression (New)	UNIF(4,6) minutes

Station Inventory — Use the Station module (Advanced Transfer)



Name	Station Inventory
Station Type	Station
Station Name	s Inventory

Route to Paint — Use the Route module



Name	Route to Paint
Route Time	2 Minutes
Destination Type	Station
Station Name	s Paint

Station Paint — Use the Station module

Name	Station Paint
Station Type	Station
Station Name	s Paint

Route to Inspection — Use the Route module

Name	Route to Inspection
Route Time	3 Minutes
Destination Type	Station
Station Name	s Inspection

Station Inspection — Use the Station module

Name	Station Inspection
Station Type	Station
Station Name	s Inspection

Route to Shipping — Use the Route module

Name	Route to Shipping
Route Time	3 Minutes
Destination Type	Station
Station Name	s Shipping

Station Shipping — Use the Station module

Name	Station Shipping
Station Type	Station
Station Name	s Shipping

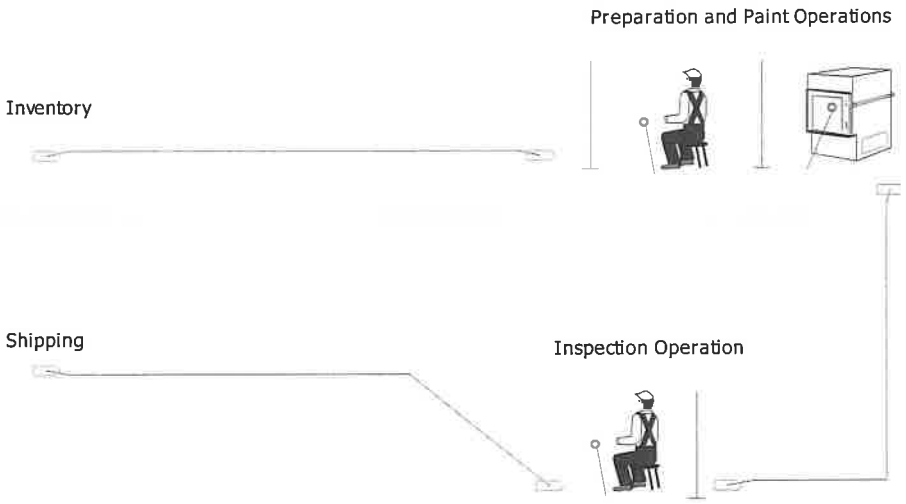
Run Setup

Number of Replications	1
Replication Length	5 Days
Hours Per Day	8
Base Time Units	Minutes

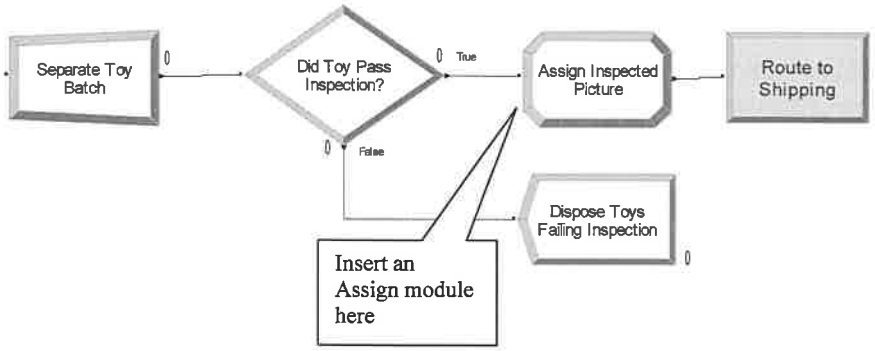
Toy Paint Shop with Routes Animation

After you have tested that your model is running properly, add animation that will help others understand the functions of the model. In this next example, we will learn how to implement the following animation concepts:

1. Create custom entity pictures.
2. Update the entity pictures in the animation for graphical representation of different stages of processing.
3. Animate the movement of entities from one station to another.



Update Entity Picture Inspection Decision

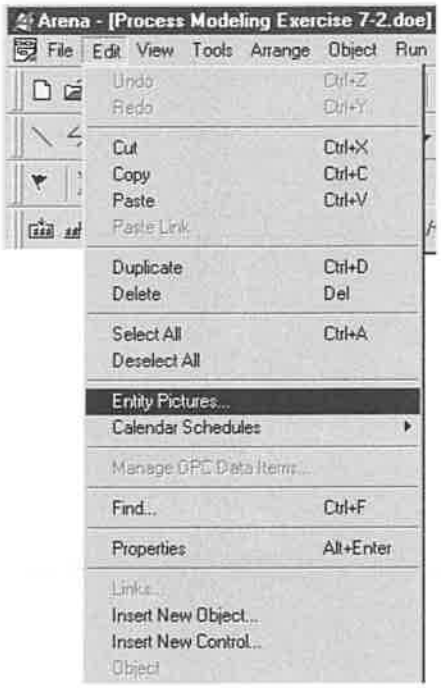


Assign New Entity Picture — Use the Assign module (Basic Process)

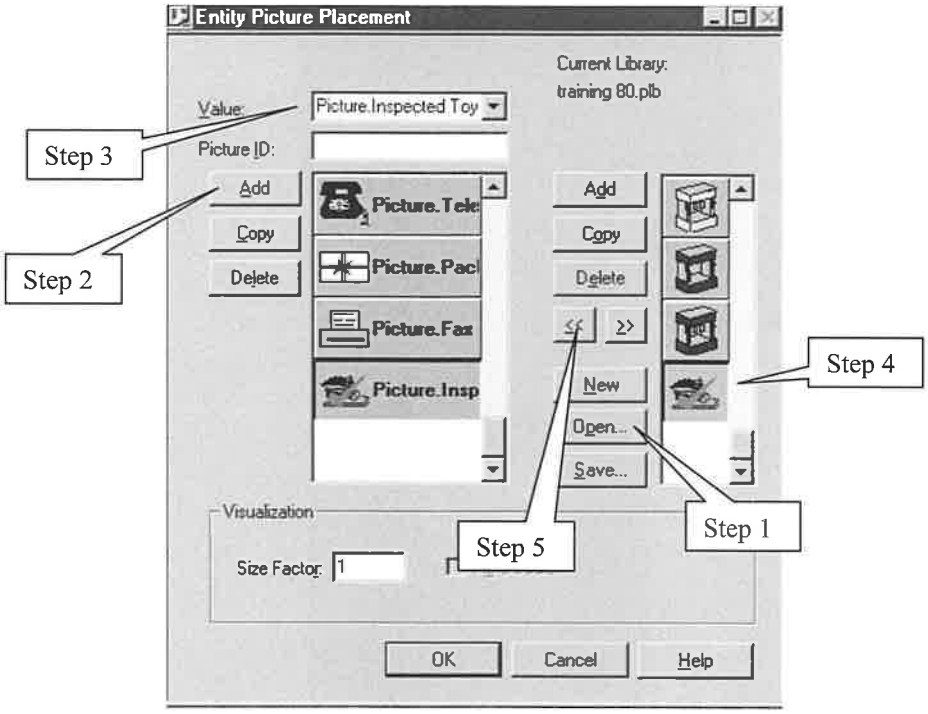


Name	Assign Inspected Picture
Type	Entity Picture
Entity Picture	Picture.Inspected Toy

Create Custom Entity Pictures — Select **Edit > Entity Pictures** from the menu bar



Creating Custom Entity Pictures



Step 1	Open the "training 80.plb" picture library
Step 2	"Add" a new entity picture
Step 3	In the Value field, Name the new picture: "Picture.Inspected Toy"
Step 4	Select the picture from the picture library that you would like to use
Step 5	Transfer the picture from the library to the new picture

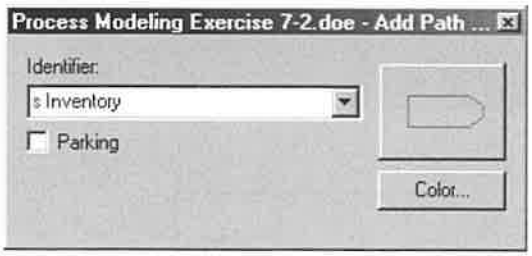
NOTE: The name of the entity picture does not have to contain the prefix "Picture."

Animating Stations and Routes — Attach the **Animate Transfer** toolbar



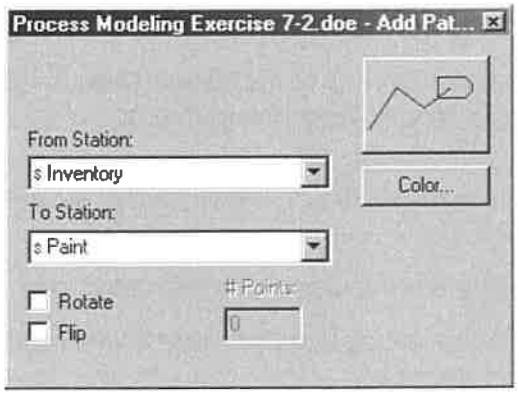
Animate Station	
Animate Route	

Animate Inventory Station Node



Step 1	Select the **Animate Station** node from the **Animate Transfer** toolbar
Step 2	Use the drop-down menu to select the Station to animate (s Inventory)
Step 3	Click **OK** after choosing the desired station
Step 4	Cross hairs should appear in the model window. Left-mouse click to place the station in the model window in its appropriate location.
Step 5	Repeat for remaining stations to be animated.

Connect the appropriate stations using the Animate node — Click **OK** and click on the station nodes that are being routed to and from.



Step 1	Select the **Animate Route** node from the **Animate Transfer** toolbar
Step 2	Click **OK** after the desired routing specifications are chosen.
Step 3	Cross hairs should appear in the model window. Left-mouse click in the originating station involved in the routing.
Step 4	Cross hairs should remain, but there should also be a line segment dragging with the cross hairs. Now, select by a left-click in the station the target station to which the entity will be traveling.
Step 5	Repeat for the remaining stations that are routed between one another.

NOTE: When placing routing segments, it is not necessary to worry about first connecting the starting location to the destination station.

Move the Seize Queues and the Resource Animation and create an Animation View of the Toy Paint Shop separate from the flow of logic.

Named Views

Select an item or group of items from the flowchart view of the model window and save the selection by providing a name and optional hot key.

To add a named view to a model, click on the **Named Views** button () from the View Toolbar or select **View > Named Views** from the menu.

When a named view is added to a model, its name and hot key appear on the Project Bar's Navigate panel.

Each submodel view gets its own independent set of named views.

To zoom to a particular view, simply click on the desired view from the list in the navigation tree or press the hot key.

Hot-key values can be any alphanumeric character and are case sensitive.

Topics covered in this section:

Concepts

- Creating user-specified statistics
- Creating and using user-specified mathematical expressions
- Single- and two-dimensional arrays for both Variables and Expressions
- Grouping of similar elements (Sets)
- Arena tool for building complex expressions (Expression Builder)

Arena Constructs

- Record logic module
- Expression spreadsheet module
- Variable spreadsheet module
- Set spreadsheet module
- Expression Builder

Concepts and Terminology

Record Module



This module is used to collect user-defined statistics in the simulation model. Observational statistics can be collected individually or from a set. The various types of observational statistics available include:

Count — An integer value that is used for incrementing or decrementing (e.g., parts through a process, amount in storage – incremented when placed in storage, decremented when removed from storage).

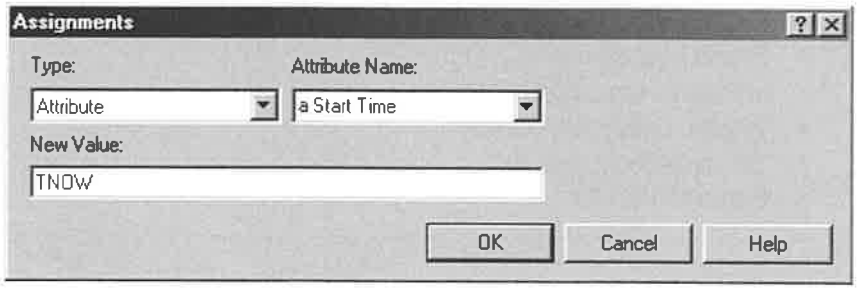
Entity Statistics — This module will record all of the entity's cost and time information. This may be valuable if the entity type will be changed using an Assign module and statistics are desired on the entity before the change. (e.g., cycle time, costing, etc.)

Time Interval — Difference between the current time and the attribute value of an entity (e.g., Time in System, Time from one process to another, etc.).

Time Between — The time between entity arrivals at the module (e.g., Time Between Arrivals, Time Between Failures, etc.)

Expression — Any specified expression.

NOTE: Often when using a Time Interval statistic, an attribute is placed on the entity for the purpose of a time-stamp (allowing for the time interval to be calculated). When a time-stamp is used, the attribute is set equal to the Arena reserved word *TNOW*. *TNOW* is equivalent to the current simulation clock time.



Expressions

Expressions refer to a user-named element that can contain a mathematical expression. Expressions can contain any math function available in Arena, including random values, distributions, or constant values. Expressions cannot be changed during a model run.

An expression can be defined as a single value, a single-dimension array, or as a two-dimensional array.

Typical use of an arrayed expression:

Use in a Delay module where the expression holds different mathematical distributions representing the processing time for multiple part types; each of these parts use the same resource and pass through the same delay module.

Using a single-arrayed expression to contain the varying distributions, the expression can be called in the Delay module via "ExpressionName(Part Type)." "ExpressionName" refers to the name defined in the Expressions module and "Part Type" refers to an attribute value assigned to each part defining the parts number.

Arrayed Variables

User-defined variables can be arrayed to provide single- or double-dimension arrays. Variables can come in the form of integer or real values. Variables can be assigned new values during a model run.

Variables can be used for anything ranging from incrementing and decrementing counters to storing unique part attributes to holding numerical values for expressions.

One important function of Variables is its ability to hold values read in from outside sources. This functionality increases model flexibility. It is important to remember that array sizes must be set prior to model runs because, once set, array size cannot be changed. Keep in mind that all values in the array do not have to be used (i.e., array values can be zero).

Variable default values are set at zero, although initial values can be set (modifying Variables module or read in from flat file(s)).

Sets

Sets group similar elements so that they can be referenced via a common name and set index.

Typical sets would contain groups of resources, queues, stations, pictures, counters, tallies, or expressions. Individual sets can only contain elements of the same type; mixed elements cannot be contained in a set.

Objects in a set can be referenced directly or via the set name and index.

Process Modeling Exercise 3 (Process Modeling Exercise 7-3.doe)

Based on the success of his first two simulation models, the manager at the Toy Paint Shop has decided to add even greater detail to the model. The manager has requested that different processing times for the two toys be used in the preparation process and that the model use different inspection pass rates for each toy type.

In the original model, both toy types required the same amount of preparation time and both toys had the same inspection pass rate. In reality, the preparation and the inspection rate are dependent on the type of part being processed. An intern was assigned to perform a time study on the preparation process and found that the actual process times for each part type follow the distributions shown below.

Toy Boat TRIA(2, 3, 5) minutes

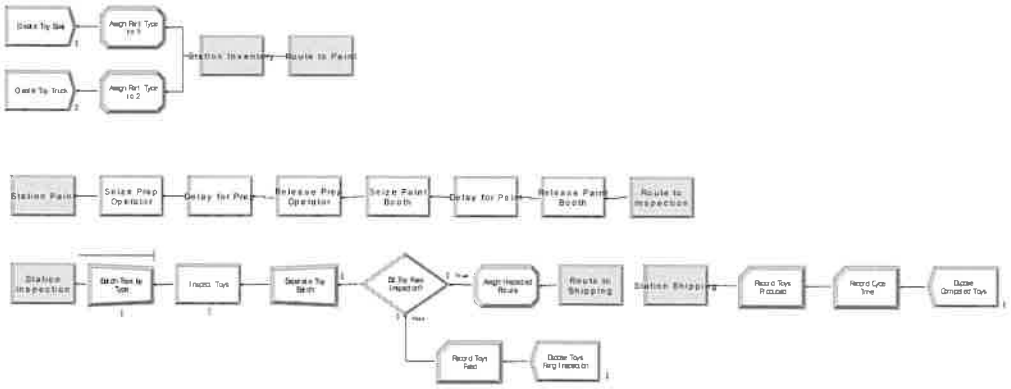
Toy Truck TRIA(4, 5, 8) minutes

The intern also found that the inspection pass rate for boats was 95% and the rate for trucks was 85%.

In addition to these changes, the manager would like to know the average cycle time for all toys passing inspection and wants the model to track the number of toys passing and failing inspection by part type.

Run the model for five 8-hour days. The base time units should be minutes.

Examine part cycle-time statistics, number in queue statistics, resource utilization statistics, and the user-defined statistics.



Modeling Approach

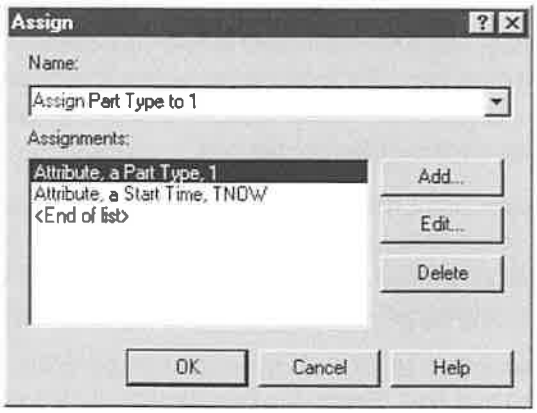
Toy Cycle Time Tally: Add a Record module before the Dispose Complete Toys to tally the time interval between the part's arrival in the system and the part's exit of the system. Use a new attribute Start Time to store the arrival time. Update the Assign Part Type modules to mark the arrival time for the part.

Toy Pass/Fail Counts: Add a Record module before the Record Cycle Time and another before the Dispose Toys Failing Inspection to count the number of completed parts into a set. Use the part type as an index. Remember that you will need to establish a set of counters for the toys passing the inspection and another for the toys failing.

Unique Prep Times: Set up an arrayed Expression containing the new prep time distributions for each part type. Change the prep Delay module to reference the expression with the part type as the index.

Unique Inspection Pass Rates: Set up an arrayed Variable containing the new inspection pass rates for each part type. Change the Percent True in the Decide module to reference the arrayed variable with the part type as the index.

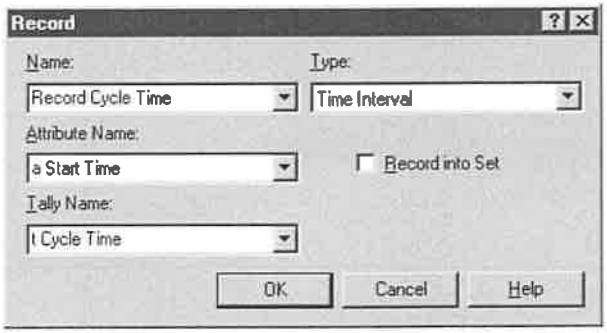
Assign Start Time — Update the Assign module



Name	Assign Part Type to 1
Assignment Type	Attribute
Attribute	a Start Time
Value	TNOW (current simulation time)

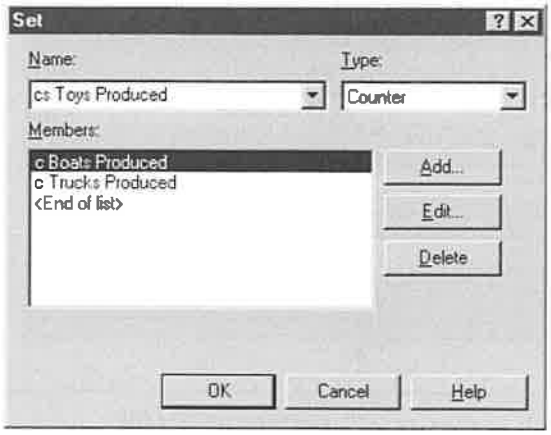
Do this for the second Assign module.

Record Cycle Time — Use the Record module



Name	Record Cycle Time
Type	Time Interval
Attribute Name	a Start Time
Tally Name	t Cycle Time

Count Toys Passing Inspection by Type — Use the Set spreadsheet

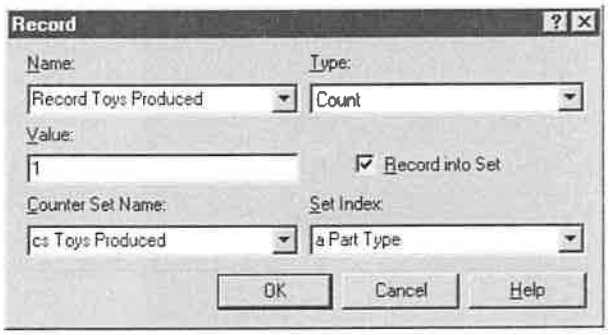


Name	cs Toys Produced
Type	Counter
Members	2 rows
Counter Name	c Boats Produced
Counter Name	c Trucks Produced

Count Toys Failing Inspection by Type

Name	cs Toys Failed
Type	Counter
Members	2 rows
Counter Name	c Boats Failed
Counter Name	c Trucks Failed

Record Toys Produced — Use the Record module

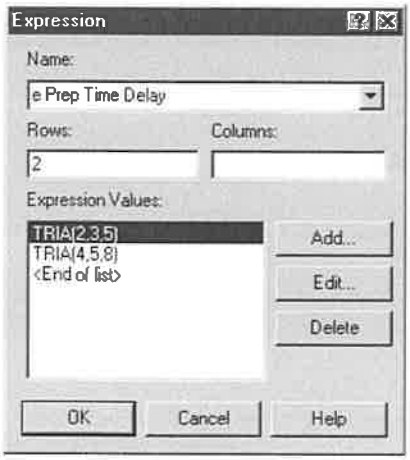


Name	Record Toys Produced
Type	Count
Value	1
Record into Set	*Check*
Counter Set Name	cs Toys Produced
Set Index	a Part Type

Record Toys Failed — Use the Record module

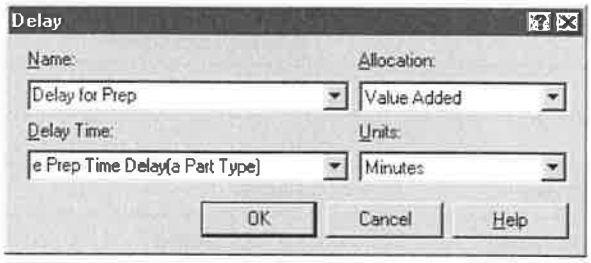
Name	Record Toys Failed
Type	Count
Value	1
Record into Set	*Check*
Counter Set Name	cs Toys Failed
Set Index	a Part Type

Set up the Prep Time — Use the Expression spreadsheet



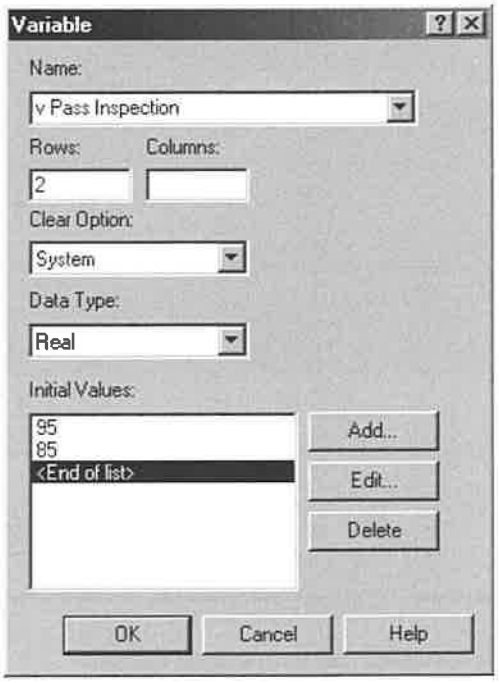
Name	e Prep Time Delay
Rows	2
Expression Values	TRIA(2, 3, 5)
Expression Values	TRIA(4, 5, 8)

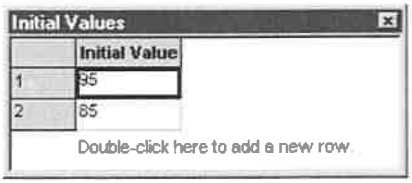
Delay for Prep — Modify the Delay module



Name	Delay for Prep
Delay Time	e Prep Time Delay(a Part Type)

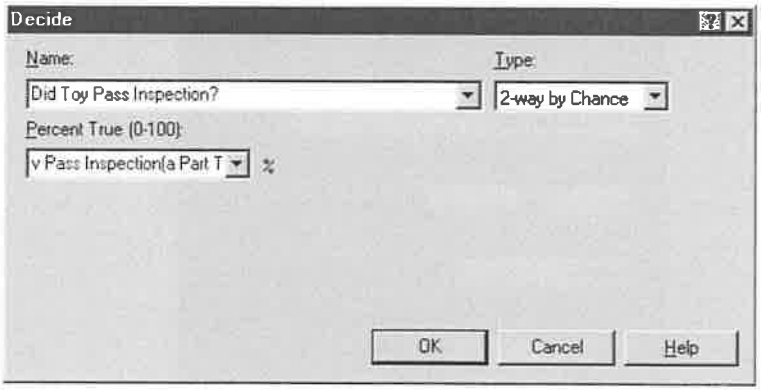
Set up the Passing Inspection Rates — Use the Variable spreadsheet





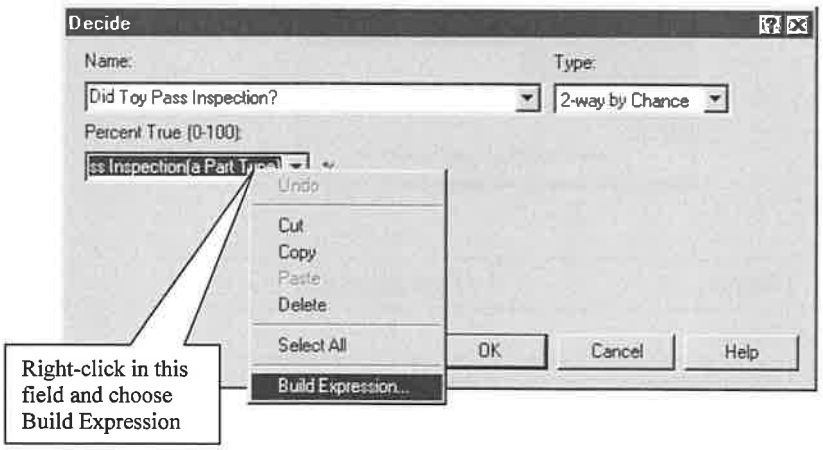
Name	v Pass Inspection
Rows	2
Clear Option	System
Initial Values	95
Initial Values	85

Did Toy Pass Inspection? — Modify the Decide module

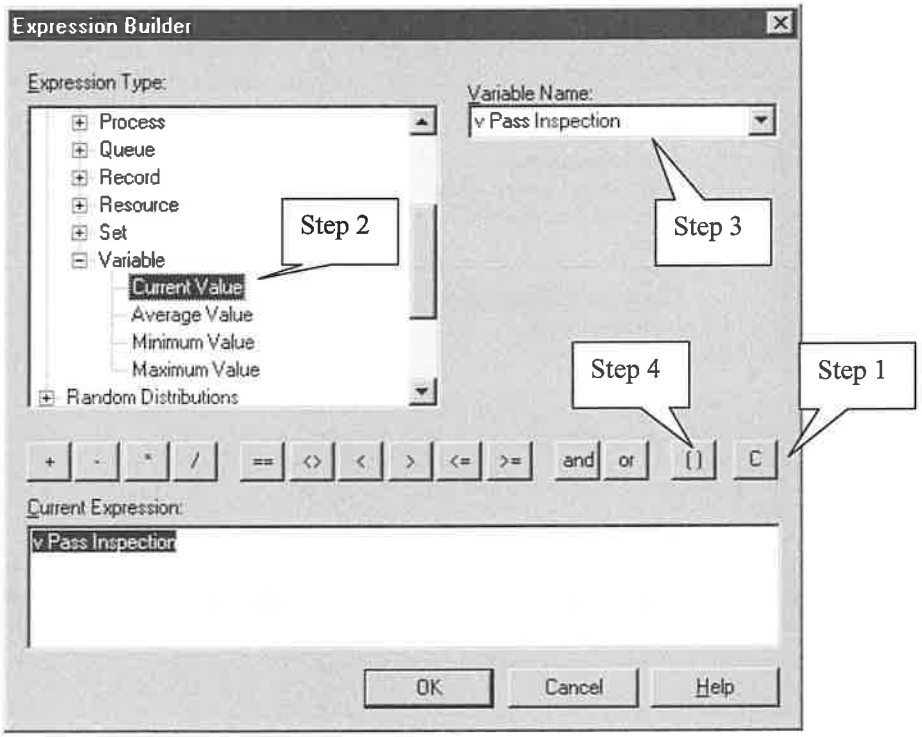


Name	Did Toy Pass Inspection?
Percent True	v Pass Inspection(a Part Type)

Build Chance Condition Using the Expression Editor

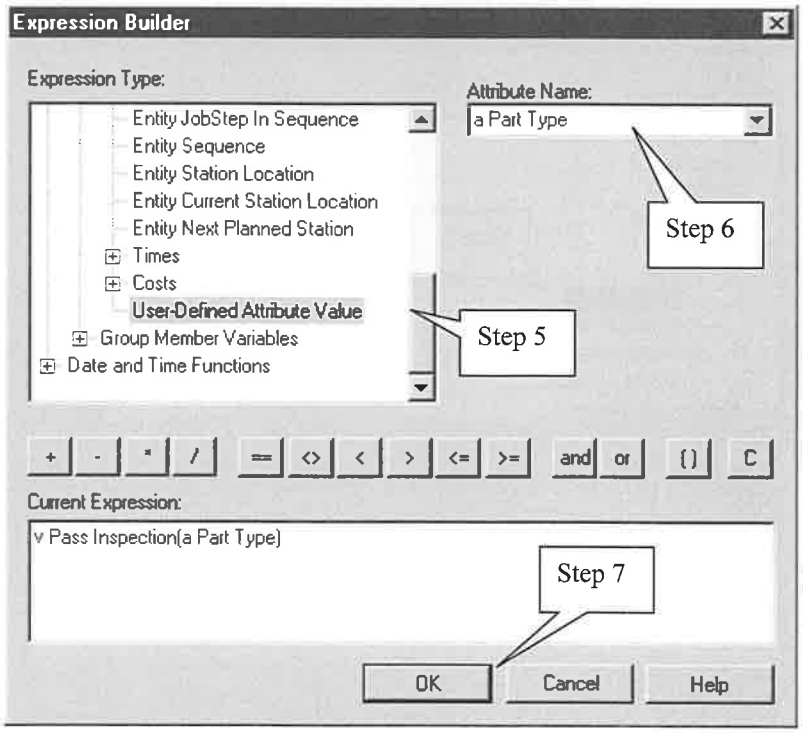


Build the Expression



Step 1	Clear Expression C
Step 2	Choose the variable located in **"Basic Process Variables\Variable\Current Value"**
Step 3	Choose the variable required for the expression: **"v Pass Inspection"**
Step 4	Add a set of parentheses ()

Build the Expression (cont.)



Step 5	Choose the index for the variable located in **"Entity-Related Variables\Attributes\User-Defined Attribute Value"**
Step 6	Choose the attribute required for the expression: **"a Part Type"**
Step 7	Click **OK** to exit.

Topics covered in this section:

Concepts

- Restricting queue size (overlapping Resources)
- Pull systems

Arena Constructs

- Resource spreadsheet module

Concepts and Terminology

Overlapping Resources

Many systems have limited buffers or no buffers between operations, which can cause the previous resource to be blocked. In these cases, an entity retains control of the resource until it can move to the next operation or into the buffer.

A common way to model these systems is to use the concept of overlapping resources.

If there is no buffer between operations, the entity retains control of the first resource until it can seize control of the following resource. Only then is the first resource released so that the next entity can begin processing.

If there is a limited buffer between operations, a buffer resource is defined with a capacity equal to the buffer size. The entity retains control of the first resource until there is space in the buffer (it can seize one unit of the buffer resource). It retains control of the buffer resource until it can seize the resource at the following operation, at which time it releases the buffer resource before it begins processing.

Process Modeling Exercise 4 (Process Modeling Exercise 7-4.doe)

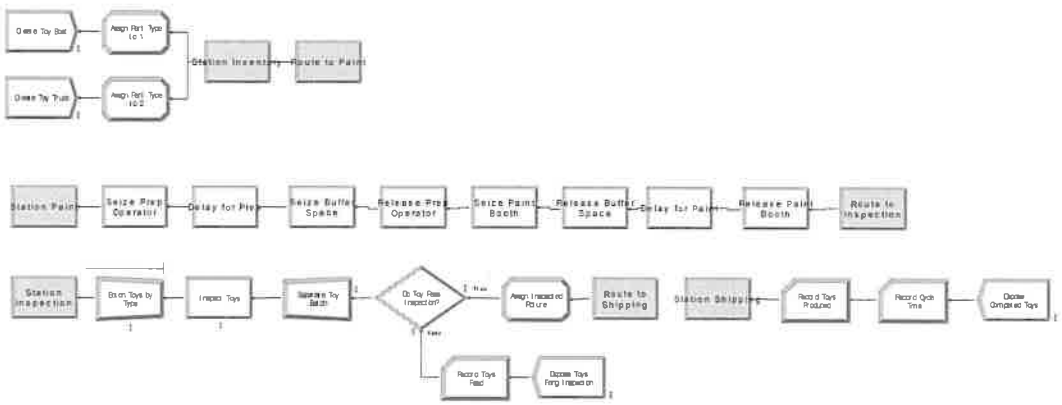
The simulation bug has bitten the manager of the Toy Paint Shop, and the manager is interested in testing a new idea.

Instead of allowing an infinite supply of work-in-process to stack up in front of the paint booth, there is only space for 3 parts. The Prep Operators will place completed parts in the 3-space buffer. If a Prep Operator completes a part and all three buffer spaces are filled, the operator will hold onto the current part and will wait for space to become available.

Once the paint booth removes a part from the buffer, the Prep Operator will move the part into the space in the buffer.

Run the model for five 8-hour days. The base time units should be minutes.

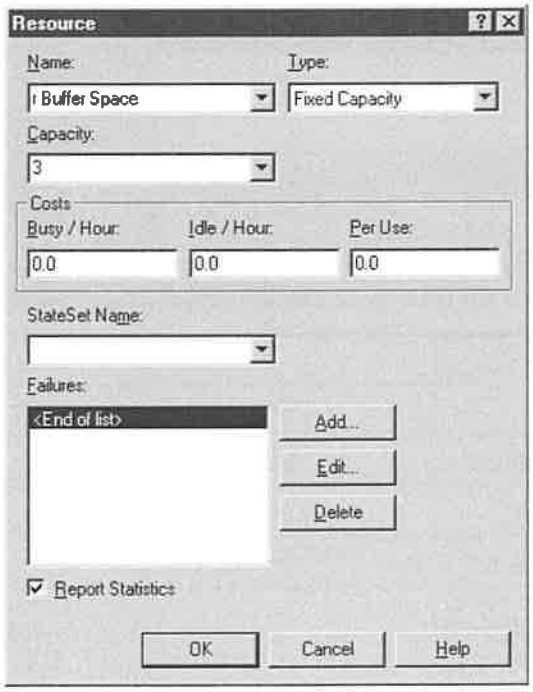
Examine part cycle-time statistics, number in queue statistics, and resource utilization statistics.



Modeling Approach

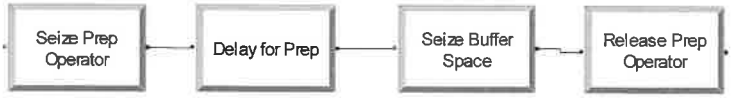
Limited Buffer Space: Use the concept of overlapping resources. Define a buffer resource with capacity of 3. Only release the Prep Operator if there is space in the buffer. Release the buffer space when a part gains control of the paint booth.

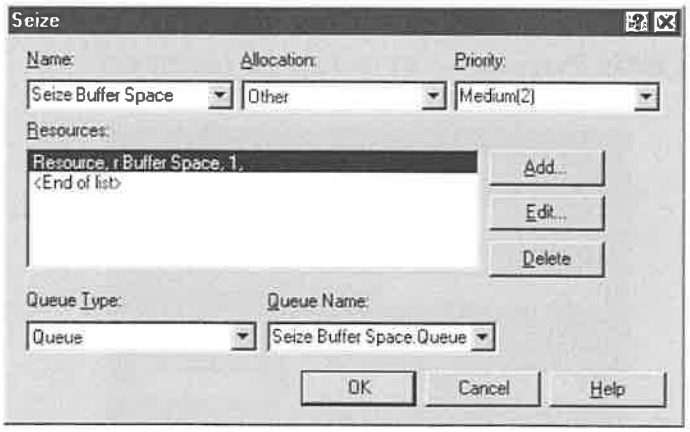
Define the Buffer Resource — Use the Resource spreadsheet



Name	r Buffer Space
Type	Fixed Capacity
Capacity	3

Overlap the Prep Operator and Buffer Resources — Add a Seize module

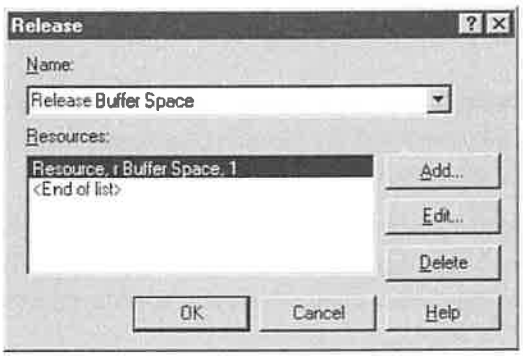




Name	Seize Buffer Space
Allocation	Other
Resource	r Buffer Space
Quantity	1

Overlap the Buffer and Paint Booth Resources — Add a Release module





Name	Release Buffer Space
Resource	r Buffer Space
Quantity	1

Process Modeling Workshop – Part 1

Two different part types enter a machine shop. Part 1 arrives exponentially with a mean of 23 minutes. Part 2 arrives exponentially with a mean of 42 minutes. These parts then route to the machining center in exactly 2 minutes.

Both part types must be processed at a single machining center with priority given to Part 2. The machining process can take anywhere from 12 to 15 minutes, but usually requires 13 minutes.

The machine center breaks down approximately every 100 minutes (EXPO(100)), and the time to repair follows a normal distribution with parameters (10,2) minutes. The failure will preempt the part being processed.

Once a part has finished machining, the different part types will follow different processing steps. Part 1 is processed in a heat-treat oven in batches of 5, while Part 2 is sent directly to QA (skipping heat-treat). Movement to either heat-treat or QA takes exactly 2 minutes.

The heat-treat oven can process only one batch at a time. The heat-treat process follows the triangular distribution with parameters (0.5, 1.0, 1.5) hours. After heat-treat, each of the parts in the batch must undergo QA separately. Movement to QA from heat-treat takes exactly 2 minutes.

Part 2's skip the heat-treat process and proceed directly to QA. However, as a result of the machining process, two identical Part 2's are created. These identical parts undergo QA individually.

There is one QA operator to perform the inspection. It takes TRIA(2,3,4) minutes to complete an inspection. The inspection time is the same for all parts and is performed on a first in, first out basis.

Once QA is complete, parts exit the system.

The machine shop runs non-stop, 24-hours a day.

Run the simulation for one 24-hour day and examine number-in-queue statistics, resource utilization statistics, and frequency statistics (for the machining center).

Base time units are minutes.

Process Modeling Workshop – Part 2

Using the previous workshop, make the following modifications:

An operational change is made so that machining is now a two-step process. There is room for only three parts between Machine Center 1 and Machine Center 2.

The processing time at Machine Center 1 takes between 3 and 4 minutes, regardless of part type. The processing time at Machine Center 2 takes between 6 to 7.5 minutes, but usually takes 6.5 minutes for Part 1. Part 2 may have a range of 7 to 8 minutes, but usually requires 7.5 minutes.

Machine Center 1 has a count-based failure for cleaning after every 100 parts. The Wait option should be specified for the decrease in resource capacity due to the cleaning failure. Cleaning takes exactly 5 minutes. Machine Center 2 retains the tool failure from Part 1 of the workshop.

Collect Frequency statistics on both machining centers.

Collect a user-defined cycle time through the entire machine shop by part type.

Update animation for the changes to the system.

8: Advanced Modeling Techniques

Topics covered in this section:

Concepts

- Movement of entities using material-handling devices (e.g., fork trucks, pallet movers, etc.)
- Animation of entity movement with material-handling devices

Arena Constructs

- Transporter spreadsheet module
- Distance spreadsheet module
- Request logic module
- Transport logic module
- Free logic module
- Animate Transfer toolbar (Distance, Transporter)

As processes and systems become more complex, it is often necessary to incorporate system constraints that cannot be simply modeled with the seizing, delaying, and releasing of resources. The system may require movement through material-handling resources like conveyors or transporters, or it may require a hold and release of entities triggered by a disjoint/separate process. Also, you may be required to interface external files for reading and writing data.

The Advanced Transfer and Advanced Process panels contain modules to help you capture the details of your material-handling systems or help you with incorporating signals from separate flows. The Advanced Process panel also contains modules and spreadsheets to support the reading and writing of data to external files.

Concepts and Terminology

Transporters

Physical devices that move the entity from one station to another.

Can be used to represent material-handling or transfer devices, such as a fork truck or personnel, whose movements are important to the modeling of the system.

Entities *request* a transporter to gain control of a transporter unit and to move that unit to the location of the requesting entity.

Entities are *transported* from one station to another with a transporter.

Entities *free* transporters when they are no longer required.

The transporter must be defined using the Transporter spreadsheet.

Free-Path Transporters

Can move freely about the system without encountering any obstructions.

The travel time depends on the distance to be traveled and the speed of the transporter.

The travel distance(s) must be defined using the Distance spreadsheet.

Distances (Advanced Transfer panel)

Define the physical distance the transporter will follow.

If only one distance between two stations is defined, the transporter will follow that path regardless of which direction it is traveling.

If two distances are defined, the transporter will take a different path depending on which direction it is traveling.

Guided Transporters

Movement is restricted to a predefined network of intersections and connecting links.

The travel time varies with the vehicle's speed parameters (velocity, acceleration, and deceleration) and congestion caused by other vehicles.

The most common example is the class of vehicles referred to as automated guided vehicles (AGV).

Intersections (Elements panel)

Intersections correspond to network nodes that can have a defined length.

Only one vehicle at a time can occupy an intersection.

Links (Elements panel)

Links define the connections between intersections.

Links are made of one or more zones.

Advanced Modeling Techniques Exercise 1 (Advanced Process Modeling Exercise 8-1.doe)

The manager responsible for the Toy Paint Shop wants to capture more closely the movement of toys from inspection to shipping. The current assumption of unconstrained movement between these two operations is not representative of the actual flow, and the manager wants to capture the activities of the forklift transporting the toys between these two areas. The manager is concerned that the single forklift responsible for carrying toys that pass inspection to the shipping department may not be sufficient for the shop. The manager has decided that the model should incorporate the speed and the carrying capacity of the forklift and has asked you to determine whether one forklift is sufficient for the Toy Paint Shop.

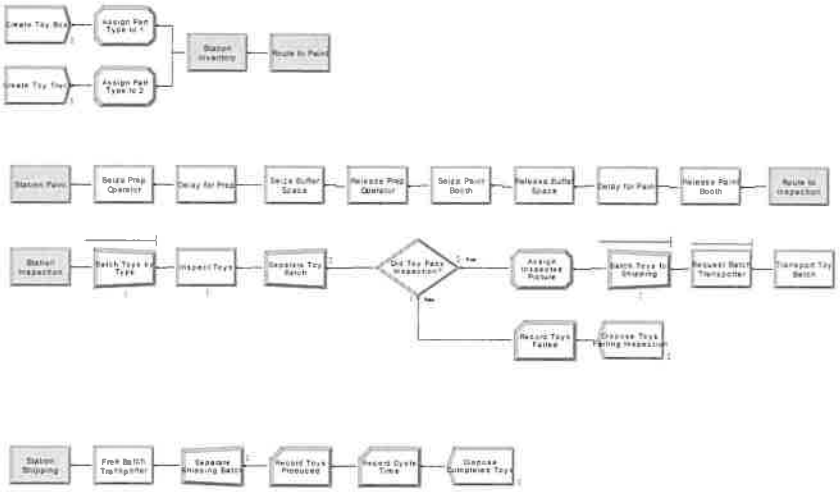
The forklift moves bins of toys from the inspection area to the shipping department at a rate of 350 feet per minute (approx. 4 mph). The forklift travels in both directions along the different aisles between stations. Distances are as follows:

From	*To*	*Distance (feet)*
Inspection Station	Shipping Station	600
Shipping Station	Inspection Station	700

The forklift is not requested to transport completed toys to the shipping department until a bin is full. The bins are designed to hold five toys, and a bin can have both boats and trucks in it at the same time.

Run the model for five 8-hour days. The base time units should be minutes.

Examine the effect of modeling the forklift explicitly on cycle time and toy throughput.



Modeling Approach

Batch Transporter (Forklift):

Use the Transporter spreadsheet to define the Batch Transporter parameters.

Use the Request module to gain control of the Batch Transporter.

Use the Transport module to move the entity from the pick-up to drop-off points.

Use the Free module to free the Batch Transporter.

Batch Transporter Paths:

Use the Distance spreadsheet to define the Batch Transporter paths for all possible trips between pairs of stations.

Animation:

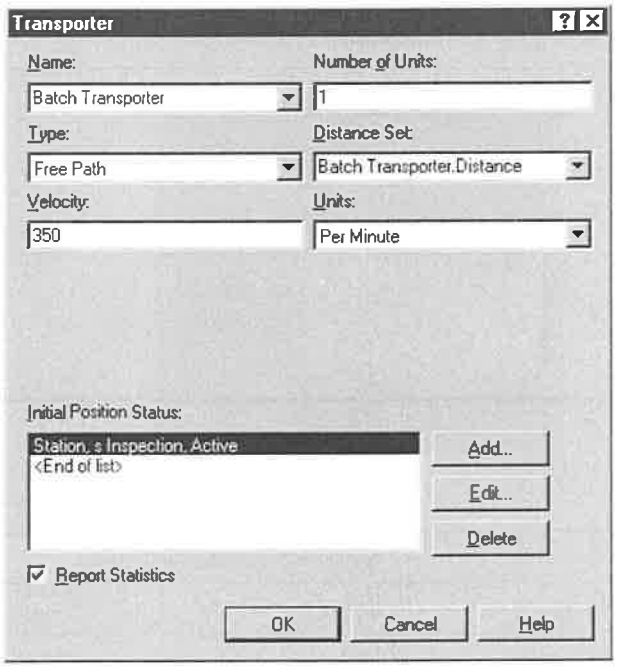
Draw Distance paths between the appropriate stations.

Add a Transporter picture and add a Ride Point to the Busy transporter picture.

Add parking areas to the stations to hold the forklift when it's not moving.

Define Batch Transporter Parameters — Use the Transporter spreadsheet





Name	Batch Transporter
Capacity	1
Type	Free Path
Distance Set	Batch Transporter.Distance
Velocity	350
Units	Per Minute
Initial Position	Station
Station Name	s Inspection

Remember: Velocity is a unit of measure per base time unit. Be consistent with your units of measure. If your distance is measured in feet, your velocity should be based on feet.

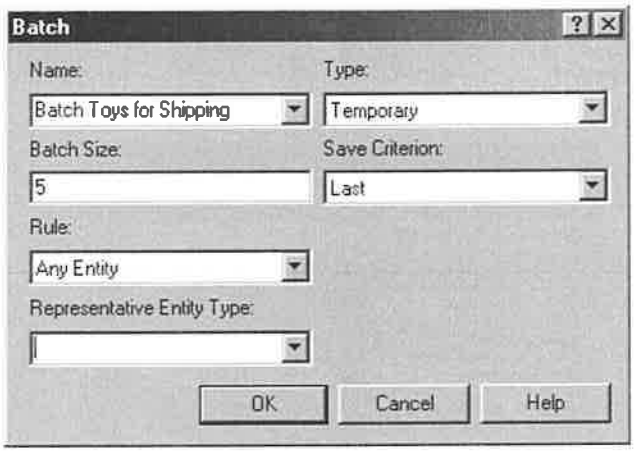
Example: Base Time Units are Minutes
Distance Units are Feet
Velocity must be expressed as Feet per Minute

Define Distances — Use the Distance spreadsheet



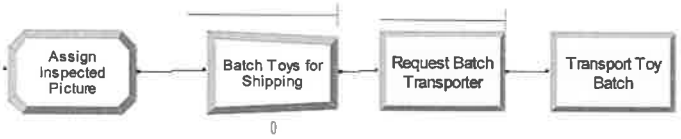
Name	Batch Transporter.Distance
s Inspection – s Shipping	600
s Shipping – s Inspection	700

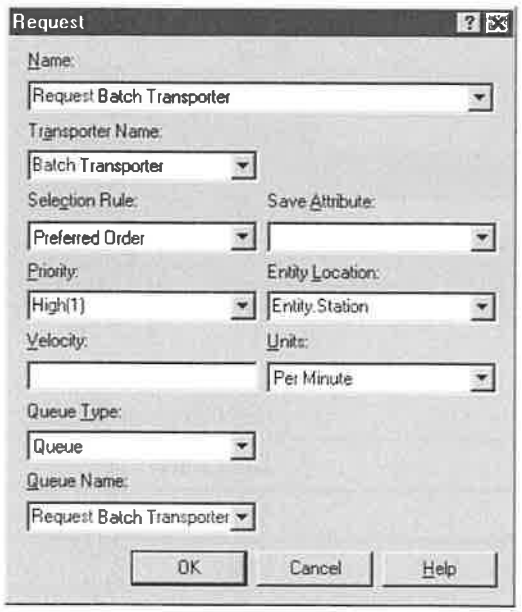
Batch Toys for Shipping — Use the Batch module



Name	Batch Toys for Shipping
Type	Temporary
Batch Size	5
Save Criterion	Last
Rule	Any Entity

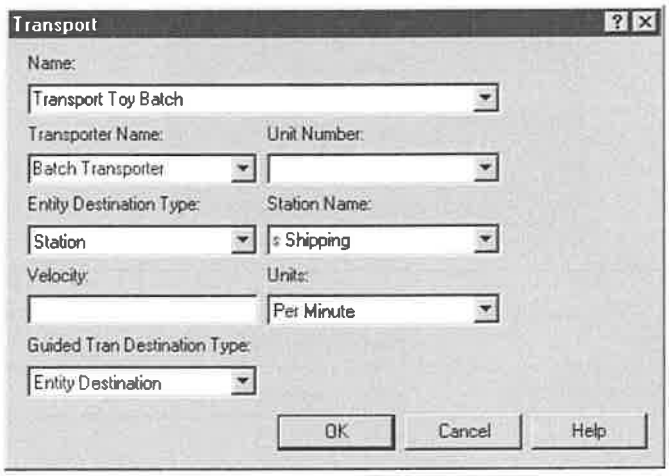
Request Batch Transporter — Use the Request module





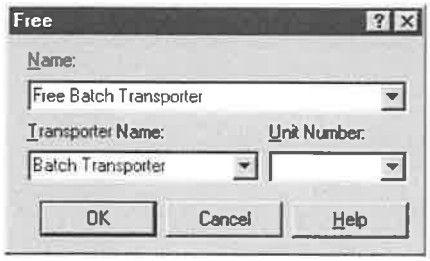
Name	Request Batch Transporter
Transporter Name	Batch Transporter
Priority	High(1)
Units	Per Minute
Selection Rule	Preferred Order
Entity Location	Entity.Station
Queue Type	Queue
Queue Name	Request Batch Transporter.Queue

Move the Entity Controlling Batch Transporter — Use the Transport module



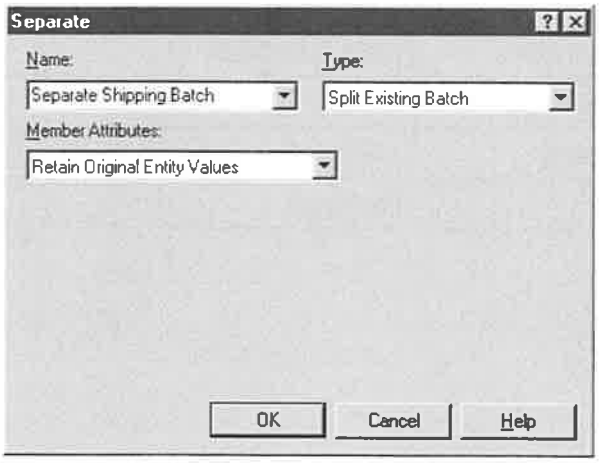
Name	Transport Toy Batch
Transporter Name	Batch Transporter
Destination Type	Station
Station Name	s Shipping
Units	Per Minute

Free the Batch Transporter — Use the Free module



Name	Free Batch Transporter
Transporter Name	Batch Transporter

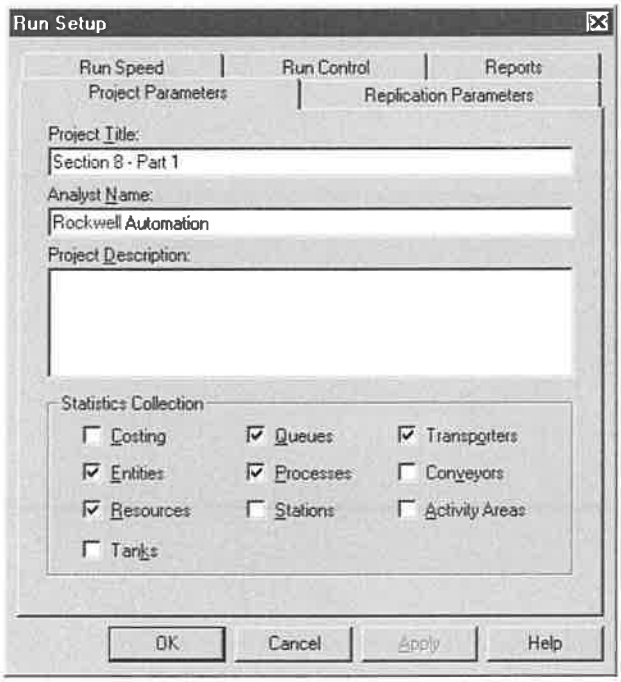
Separate Shipping Batch — Use the Separate module



Name	Separate Shipping Batch
Type	Split Existing Batch
Member Attributes	Retain Original Entity Values

Collecting Transporter Statistics

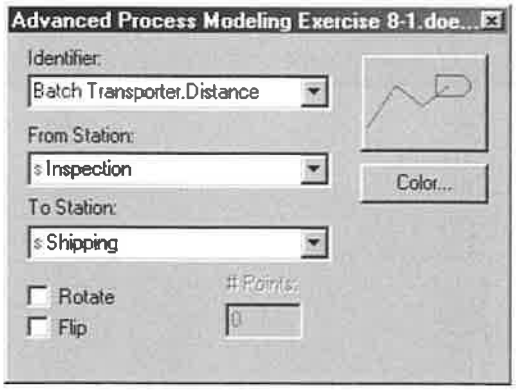
In the **Run > Setup** dialog (Project Parameters tab), click on Transporters to turn on collection of transporter statistics.



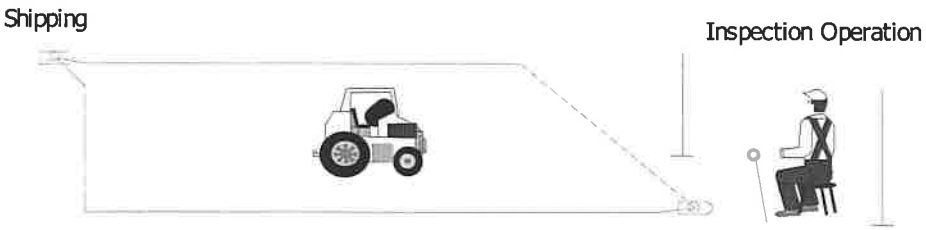
Animation

Replace the Route paths with Distance paths between the Inspection Station and the Shipping Station.

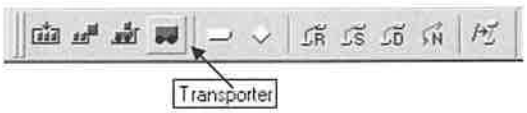


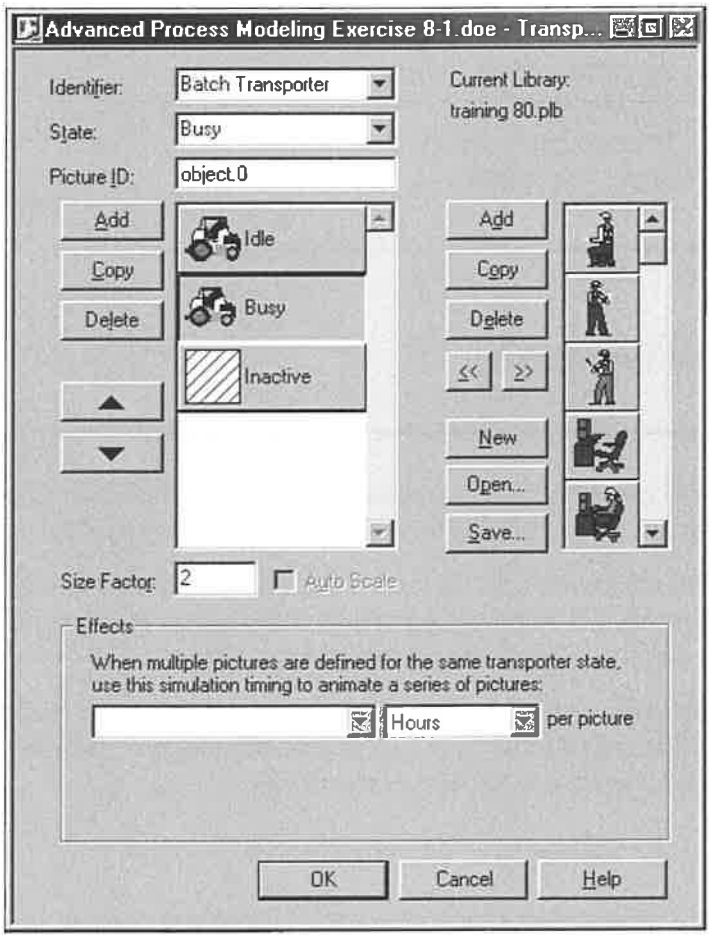


Move the Request queues (cut, paste) to the animation view.



Add a picture for the Batch Transporter





Topics covered in this section:

Concepts

- Restricting entity movement
- Model hierarchy

Arena Constructs

- Hold logic module
- Signal logic module
- Process logic module
- Object > Submodel

Concepts and Terminology

Hold

The Hold module can be used in a model to stop the movement of entities. The Hold module has two options to allow the entities to continue moving in the model:

Wait: Entities arriving at the module are held in a queue until the appropriate signal (integer value) is received. Upon receipt of the appropriate signal, the Hold module allows entities to proceed to the next section of logic.

Scan: Entities arriving at the module are held in a queue until a user-defined condition becomes true. Once the condition becomes true, the Hold module allows entities to proceed to the next section of logic.

Signal

The Signal module causes a signal to be sent to the appropriate Hold module(s) causing the Hold modules to release all, or a specified number of, entities.

NOTE: The Hold-Wait module and the Signal module are always used in combination.

Hierarchy

Arena provides the ability to segregate your Arena models formally into hierarchical views, called Submodels.

Submodels

Can contain any object supported in a model window (logic, static graphics, animation).

Can be connected to other modules, other submodels, or can stand alone.

Can be used to assist in the "drill down" modeling methodology: Start at a high level and then add detail as needed.

Model Navigation

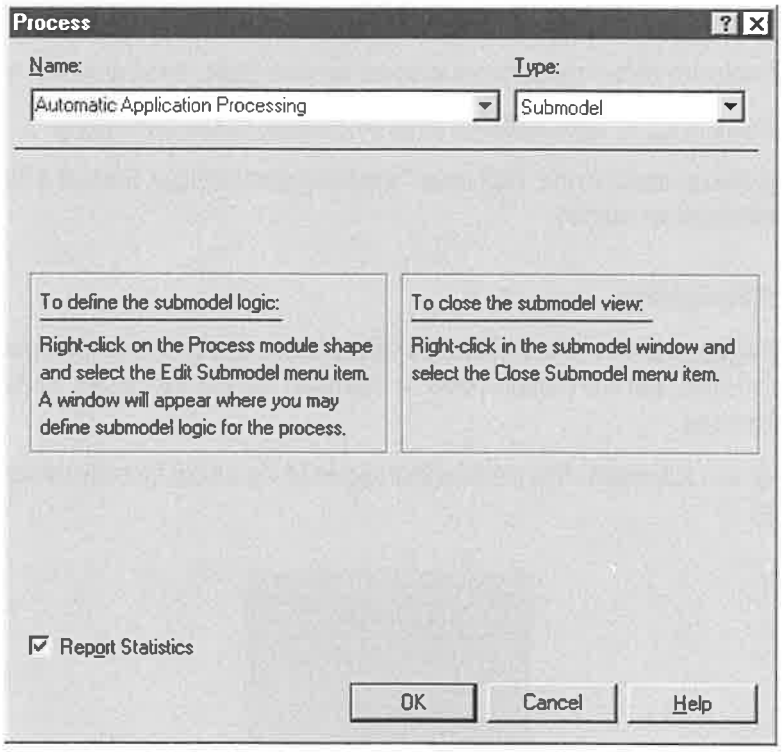
The Navigate section of the Projects bar shows a tree listing the top-level model, all of the submodels, and any Named Views. A thumbnail view of the current model level is also displayed.

Clicking on a submodel view displays that region of the model for easy navigation within a model.



Defining Model Hierarchy – Two methods

Process Submodel



Create by defining process "type" as submodel.

The Process Submodel provides the ability to aggregate a number of flowchart modules into separate submodel views, each associated with a given Process module. For an example, see *Auto Loan – Hierarchy.doe*.

Submodel Object

From the main menu, add a submodel by selecting **Object > Submodel > Add Submodel** or by clicking the **Submodel** button on the **Standard** toolbar.

This type of submodel is simply a collection of modules that have been aggregated for navigation and ease-of-use purposes.

Defining Model Hierarchy (cont.)

What is the difference between the two hierarchy methods, and which should I use?

The main difference between the two methods is the statistics generated.

When a Process module is specified as a submodel and logic is placed within the submodel view, any statistics (including costing and time information) accumulated when an entity is in the submodel will be directly reflected in the statistics for that Process. This is referred to as "rolling up" the statistics into the parent process's statistics. This is true regardless of the number of levels of hierarchy that are defined.

Statistics accumulated from logic defined within a submodel object are not aggregated.

Process Submodel and Sub-model object. Process Sub-models have all data rolled up in them, whereas Sub-model Objects have stats generated as normal.

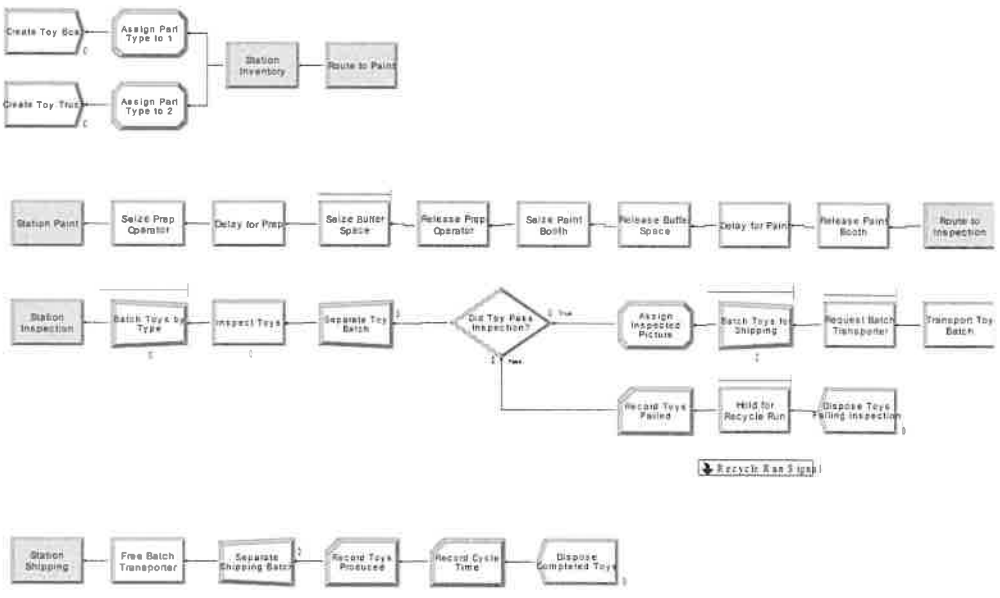
Advanced Modeling Techniques Exercise 2 (Advanced Process Modeling Exercise 8-2.doe)

In an effort to minimize waste, the management for TMDC has decided to recycle parts that fail inspection. The manager responsible for the Toy Building Shop has decided to send a cart twice a day to pick up the toys that failed inspection: 11 a.m. and 3 p.m. The Toy Paint Shop manager has decided to incorporate the holding and releasing of parts to the Toy Building Shop in the simulation.

NOTE: Assume that the Toy Paint Shop's 8-hour daily shift begins at 9 a.m. and goes until 5 p.m.

Run the model for five 8-hour days. The base time units should be minutes.

Examine the effect of modeling the recycling hold on the throughput for failed inspection parts.



Modeling Approach

Hold/Wait: Use the Hold module in the Advanced Process panel to block the movement of entities until a signal to release.

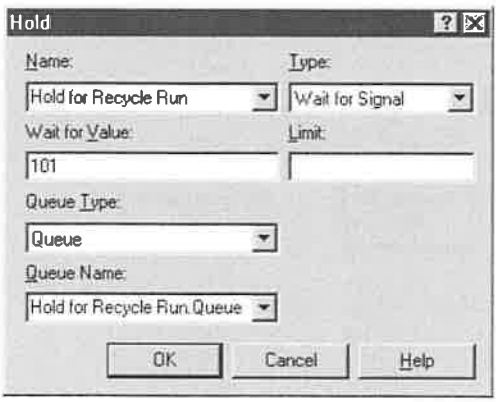
Signal: Create a Submodel with a Create module, a Signal module, and a Dispose module to signal the Hold/Wait.

Recycle Run Signal



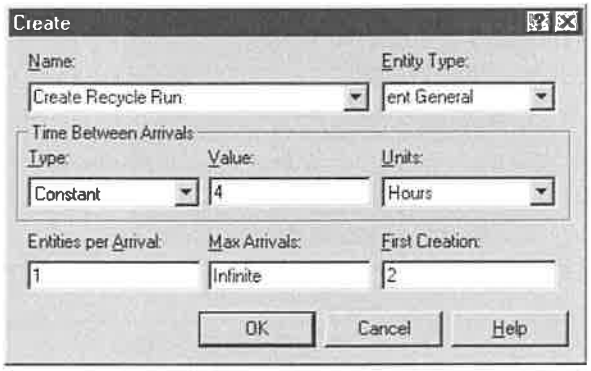
Hold for Recycle Run — Use the Hold module





Name	Hold for Recycle Run
Type	Wait for Signal
Wait for Value	101
Queue Type	Queue
Queue Name	Hold for Recycle Run.Queue

Create Recycle Run — Use the Create module



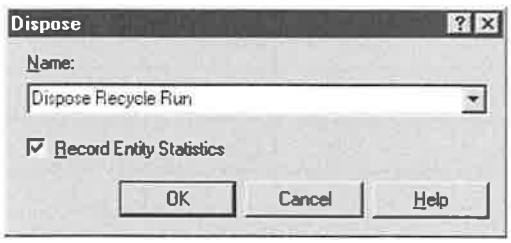
Name	Create Recycle Run
Entity Type	ent General
Type	Constant
Value	4
Units	Hours
First Creation	2

Signal Recycle Run — Use the Signal module



Name	Signal Recycle Run
Signal Value	101

Dispose Recycle Run — Use the Dispose module



Name	Dispose Recycle Run

Create Submodel

Group select the three modules for the Recycle Run by holding down the left mouse button and dragging the wireframe box to surround the modules. Then select **Object > Submodel > Aggregate** from the main menu to create the Submodel.

Topics covered in this section:

Concepts

- Reading from and writing to external data sources
- Control logic loops

Arena Constructs

- File spreadsheet module
- ReadWrite logic module

Concepts and Terminology

Reading from and Writing to an External File

The ReadWrite module gives you the ability to read or write data from or to an external source.

The Read option of the module allows you to read information in from an external file or prompt the user to enter a value from the keyboard and assign the value to an attribute or variable.

Arena can read either text (sequential) files, Lotus 123 (wks); Microsoft Excel (xls), Access (mdb), or ActiveX Data Objects (ADO); or eXtensible Markup Language (XML) files.

The ReadWrite logic varies according to the access type of the file specified.

- If sequential or Lotus, the specified file is examined to see if it is open. If it is not, the file is automatically opened. The values listed are read or written according to the format specified.
- If Microsoft Excel, Access, or ActiveX, the specified file is examined to see if an ADO connection to the file is open. If closed, an ADO connection is automatically opened to the data source. The *Recordset ID* is then examined. A recordset refers to a table of data that is organized in columns and rows. The values of the attributes, variables, or expressions listed in the *Assignments* repeat group are then read from or written to the specified *Record Number* in the recordset. The

first entry in *Assignments* is read from or written to the first field in the record, the second entry corresponds to the second field, and so on.

- If Extensible Markup Language, the specified file is examined to see if it is open. If not, the file is automatically opened into a single ADO recordset. The values listed are then read from or written to the specified *Record Number* in the recordset.

Use the Files spreadsheet to define the external file.

The Write option allows you to write out the value of an expression, variable, or attribute to an external file, the output report, or to the screen.

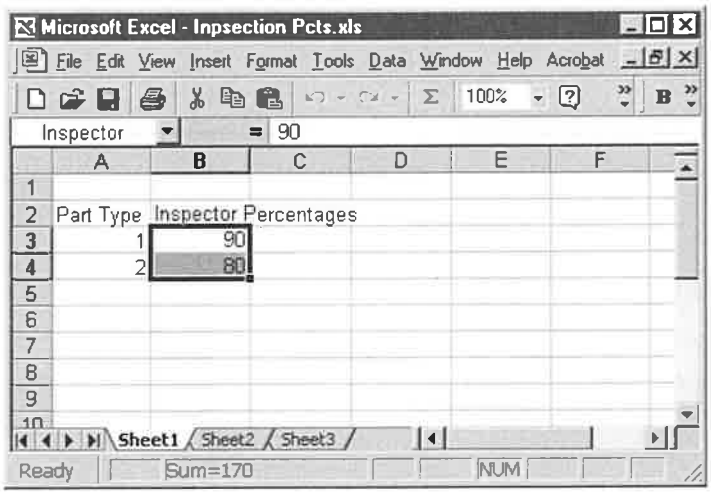
The Write option is useful for generating external files for extensive analysis and debugging.

Named Range refers to at least 2 cells.

Advanced Modeling Techniques Exercise 3 (Advanced Process Modeling Exercise 8-3.doe)

The manager from the Toy Building Shop has decided to examine the impact of recycling failed parts on the total throughput of the shop. In order to align the new requirements with the existing Toy Paint Shop simulation, the manager has requested that the Toy Paint Shop model send output data on the recycle run and part type for each part that fails inspection. The Toy Paint Shop manager has agreed and is requesting that you add this functionality to the model. Also, the intern noticed that the inspection pass rate can vary greatly based on part type, and the manager would like to have the ability to read the inspection pass rate from a Microsoft Excel spreadsheet provided by the intern.

The intern has provided an Excel spreadsheet called *Inspection Pcts.xls* in your training files directory.



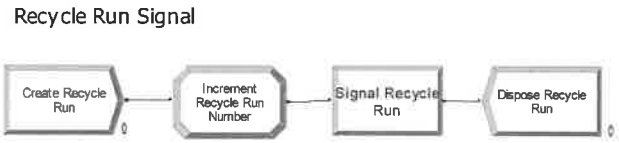
Run the model for five 8-hour days. The base time units should be minutes.

Examine the effect of the new inspection pass rates on the throughput and view the output file before handing it off to the Toy Building Shop manager.

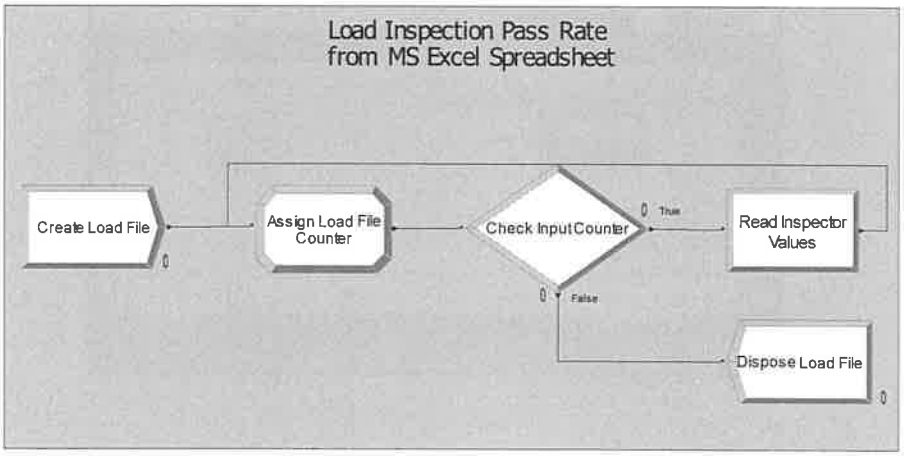
Modeling Approach

Write Data: Create a variable to track the recycle run number, and use the Assign module to increment the variable during each recycle run.

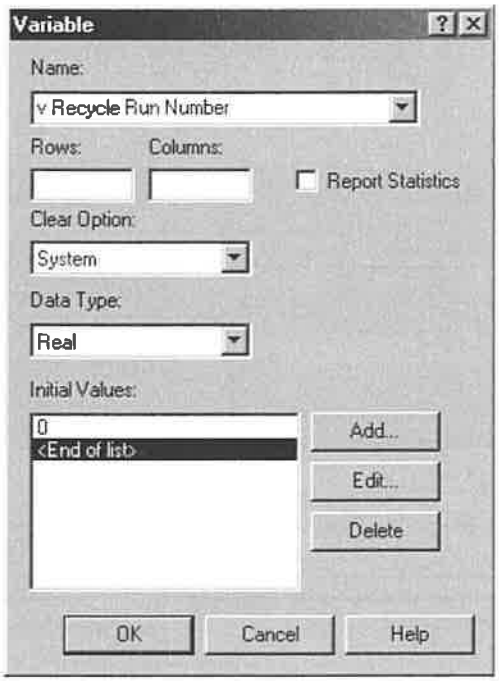
Use the ReadWrite module in the Advanced Process panel to write the recycle run number and the Part Type to a text file as toys leave the Hold module.



Read Data: Make a separate logic loop containing the ReadWrite module to read in the inspection pass rates from the *Inspection Pcts.xls* spreadsheet.

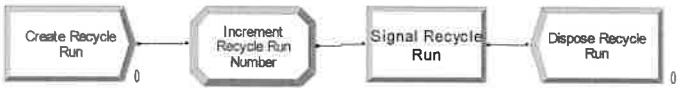


Recycle Run Number — Use the Variable spreadsheet



Name	v Recycle Run Number
Initial Values	0

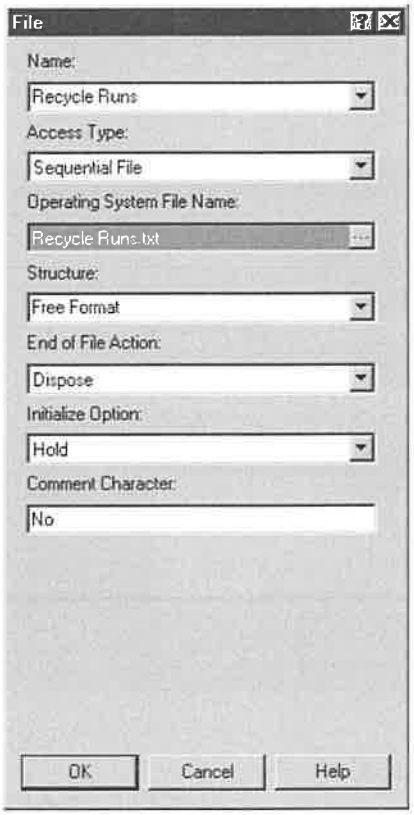
Increment Recycle Run Number — Use the Assign module





Name	Increment Recycle Run Number
Type	Variable
Variable Name	v Recycle Run Number
New Value	v Recycle Run Number + 1

File Recycle Runs — Use the File spreadsheet



Name	Recycle Runs
Access Type	Sequential File
Operating System File Name	Recycle Runs.txt
Structure	Free Format
End of File Action	Dispose
Initialize Option	Hold
Comment Character	No

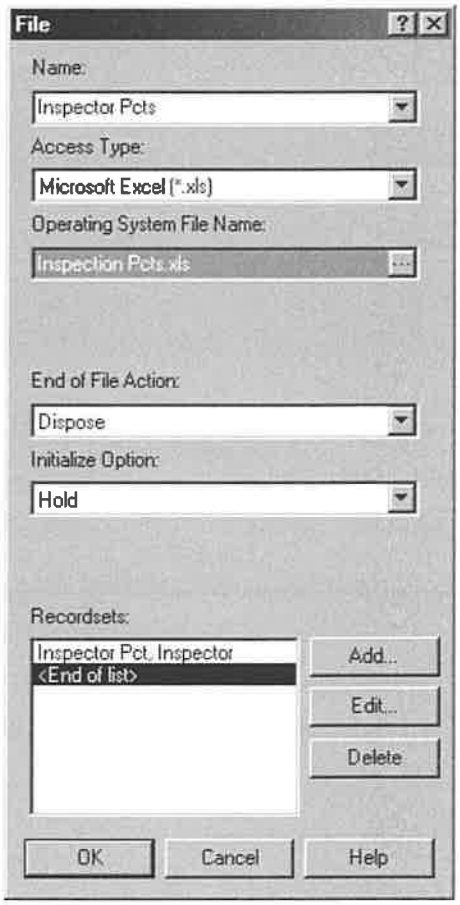
Write Recycle Run Data — Use the ReadWrite module





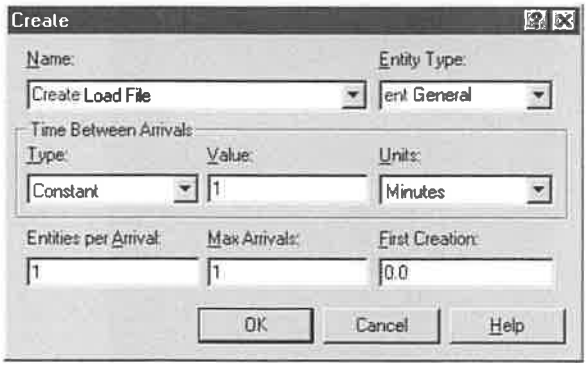
Name	Write Recycle Run Data
Type	Write to File
Arena File Name	Recycle Runs
Type	Variable
Variable Name	v Recycle Run Number
Type	Attribute
Attribute Name	a Part Type

File Inspection Pcts — Use the File spreadsheet



Name	Inspector Pcts
Access Type	Microsoft Excel (*.xls)
Operating System File Name	Inspection Pcts.xls
End of File Action	Dispose
Initialize Option	Hold
Recordset Name	Inspector Pct
Named Range	Inspector

Create Load File — Use the Create module



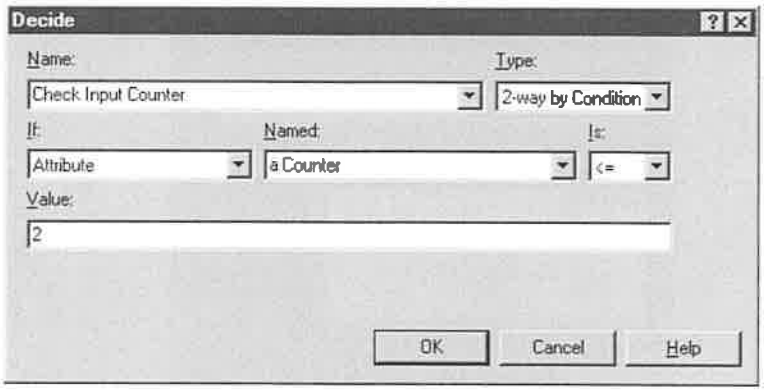
Name	Create Load File
Entity Type	ent General
Type	Constant
Value	1
Units	Minutes
Entities per Arrival	1
Max Arrivals	1
First Creation	0.0

Assign Load File Counter — Use the Assign module



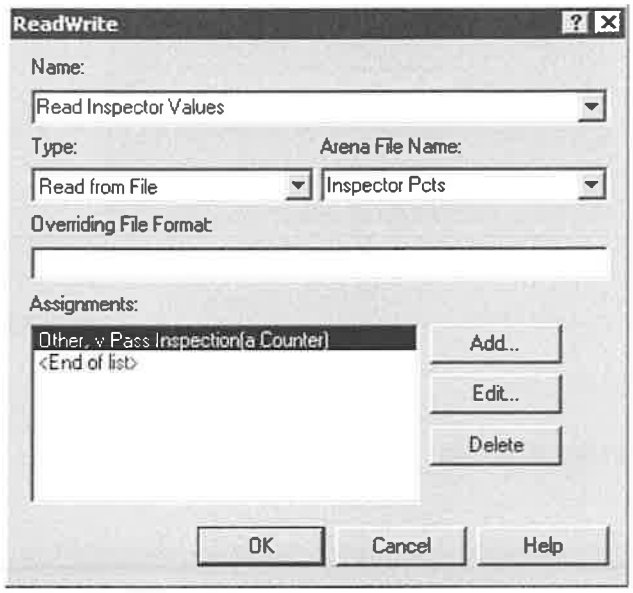
Name	Assign Load File Counter
Type	Attribute
Attribute Name	a Counter
New Value	a Counter + 1

Check Input Counter — Use the Decide module



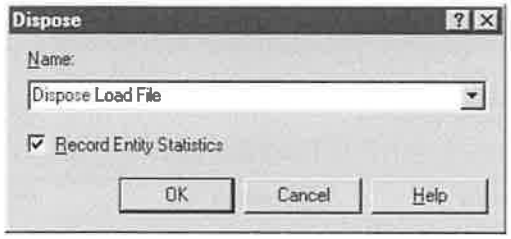
Name	Check Input Counter
Type	2-way by Condition
If	Attribute
Attribute Name	a Counter
Is	<=
Value	2

Read Inspector Values — Use the ReadWrite module



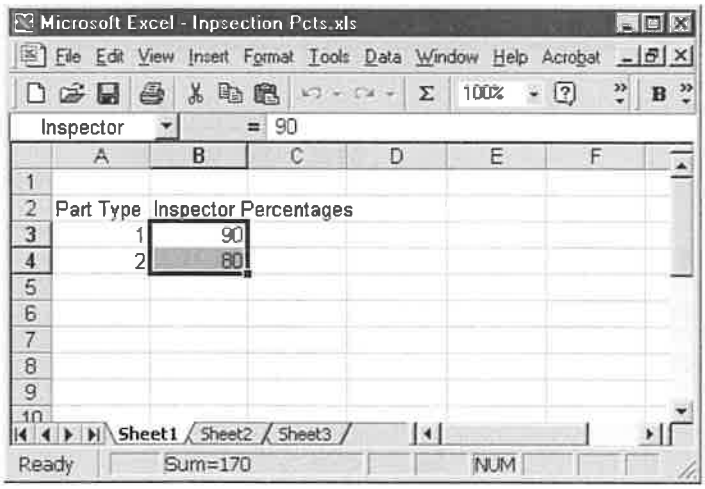
Name	Read Inspector Values
Type	Read from File
Arena File Name	Inspector Pcts
Assignments Type	Other
Assignments Other	v Pass Inspection(a Counter)

Dispose Load File — Use the Dispose module



Name	Dispose Load File

Examine Inspection Pcts File



Notice the Named Range in the file that defines the range of values to be read in. The Named Range in the Excel file corresponds to the *Recordset ID* within the Arena ReadWrite module.

The Named Range should refer to a single, contiguous block of cells.

Advanced Modeling Techniques Workshop (Advanced Modeling Workshop.doe)

Using the workshop from Section 7, make the following changes:

Transfers between the different operations within the machine shop need to be captured in greater detail. Transfers are handled by a single forklift that travels at a rate of 350 feet per minute. The forklift handles all transfers between Part Entry, the Machine Center, Heat-Treat, and QA. Each part is transferred individually. The distances between each of the areas are shown below.

From Station	*To Station*	*Distance (feet)*
Part Entry Station	Machine Center Station	700
Machine Center Station	Heat-Treat Station	420
Machine Center Station	QA Station	200
Heat-Treat Station	QA Station	220
Part Entry Station	Heat Treat Station	900
Part Entry Station	QA Station	1000

As each part leaves QA, write out the current simulation time, part type, and cycle time. Write these values to a free-format, sequential file called "EntData.txt."

HINT: Remember that cycle time is calculated from current simulation time (TNOW) – time entity entered the system.

Animate the forklift and transfers between the different station locations.

Is one forklift enough to handle the volume of parts in the machine shop?

A: User Interface

The Arena Environment

Microsoft Windows 98, Windows Me, Windows NT 4.0 (Service Pack 6a or later), Windows 2000, Windows Server 2003, or Windows XP (Service Pack 1 or later).

Drag-and-Drop Interface

Customizable Toolbars

Standard

Draw

Animate

Arrange

Animate Transfer

View

etc.

Modeling Templates

SIMAN template: Blocks and Elements

Arena template: Basic Process, Advanced Process, Advanced Transfer

Standard Toolbar



File Options

New, Open, Save

Template Options

Template Attach, Template Detach

Print Options

Print, Print Preview

Clipboard Functions

Cut, Copy, Paste

Undo/Redo Functions

Toggle Split Screen

View Region

Zoom Level

Layers Function

Submodels

Connect Function

Edit Menu Functions

Edit Time Patterns, Edit Exceptions, Display Composite View

Run Functions

Go, Step, Fast Forward, Pause, Start Over, End

Animation Speed Slider Bar

Context-sensitive Help

Help

Help Menu

Arena Help Topics — Contents, Index, Search

Using help

Modeling template help

Welcome screen access: Introduction, Tip of the Day, About Arena

Context-sensitive Help



Button to open help on menus, toolbar items, other user interface components

Help Access from Dialogs

Help action button in dialogs (e.g., modules, animation options)

What's This? Help

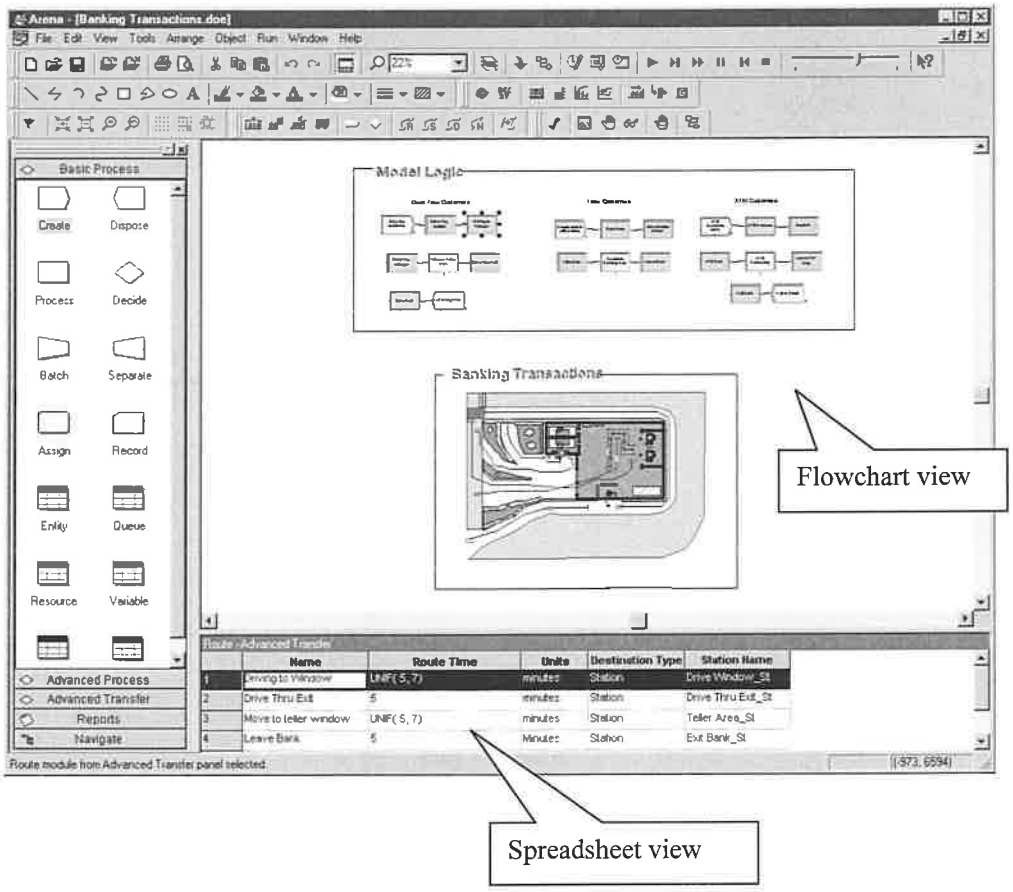


Opens help for any prompt in a dialog. Simply click on the What's This? Help button in the upper right-hand corner of the dialog and then click on the field you would like information about. The definition will appear in a pop-up window. Click again and the pop-up will disappear.

The Model Window

Flowchart view — contains all of your model graphics, including the process flowchart, animation, and other drawing elements.

Spreadsheet view — displays model data, such as times, costs, and other parameters.



The Project Bar

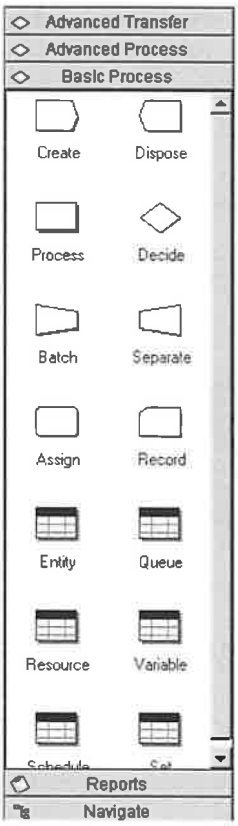
Basic Process panel — contains the modules used to define your process.

Advanced Process panel — contains the modules that allow you to capture a greater level of detail with your simulation; greater user control of entity flow.

Advanced Transfer panel — contains the modules used to model entity movement. This includes the ability to model constrained transfer devices such as transporters and conveyors.

Reports panel — contains the reports available for displaying the results of the simulation run.

Navigate panel — allows you to display different views of your model.



Model File

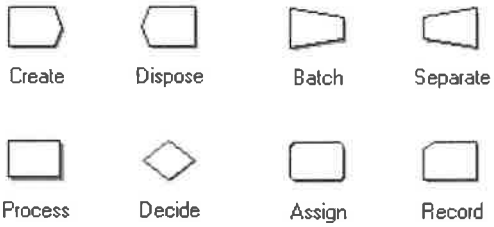
Contains all of the model definition (including the flowchart), other graphics you've drawn, and the module data you entered.

Building a Model with Modules

Flowchart Modules

These shapes are placed in the model window and connected to form a flowchart, describing the logic of your process.

These are the objects the entity will "move through," which describes the system process.



Data Modules

These are not placed in the model window. Instead, they are edited via a dialog, or the spreadsheet interface.



Mouse Actions in Arena Windows

Left Button

Click	Selects an item or object
Double-click	Opens dialog box
Ctrl+click	Extends a selection set
Shift+click	Cycles through objects
Hold and drag	Moves an object

Right Button

Click	Brings up context-sensitive menus providing a shortcut to certain functions (i.e., repeat last action, paste, undo, etc.)

Standard Keyboard Actions

F1	Opens Arena help contents
Ctrl+N	Creates new window
Ctrl+O	Opens existing file
Ctrl+S	Saves file
Ctrl+P	Prints active window
Ctrl+Tab	Cycles through open windows
Ctrl+C	Copy
Ctrl+X	Cut
Ctrl+V	Paste
Ctrl+D	Inside list — box duplicate
Ctrl+Cursor	Inside repeat groups — changes the order

Keyboard Actions in Arena Windows

Arrows	Pan left, right, up, down
PgUp / PgDn	Pan window one page up / down
Ctrl+PgUp / PgDn	Pan window one page left / right
+ / -	Zoom view in / out
Enter	Open the dialog for the selected object

Using Toolbars

Moving a Toolbar

Point to the edge or separator bar on a toolbar, or point to the toolbar title bar if it is a floating toolbar.

Drag the toolbar to a new location. If you drag it to the edge of the application window, the toolbar will dock automatically to the edge of the window.

Automatically Docking / Floating a Toolbar

Double-click on the toolbar title bar (or along an edge) to restore a toolbar to its previous position.

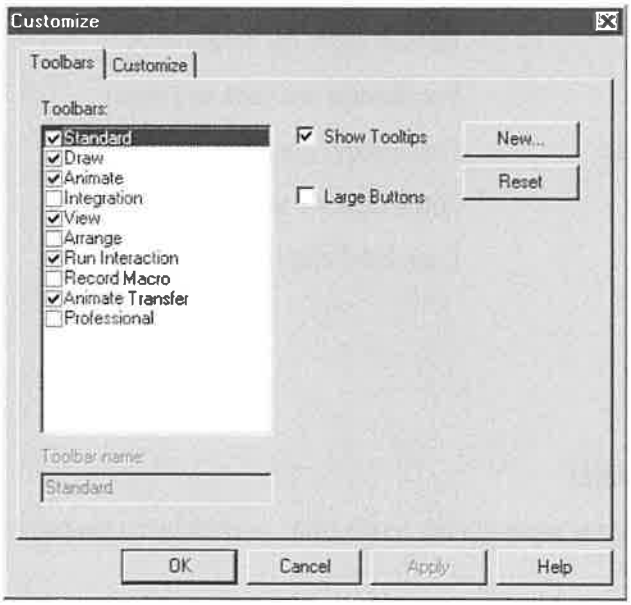
Hiding and Redisplaying Toolbars

Click the right mouse button on any visible toolbar to display the toolbar menu. Then click on the desired toolbar name to toggle its display option.

Tool Tips

Move cursor over a toolbar button and pause to display button description.

Other Options (**View > Toolbars** menu item)



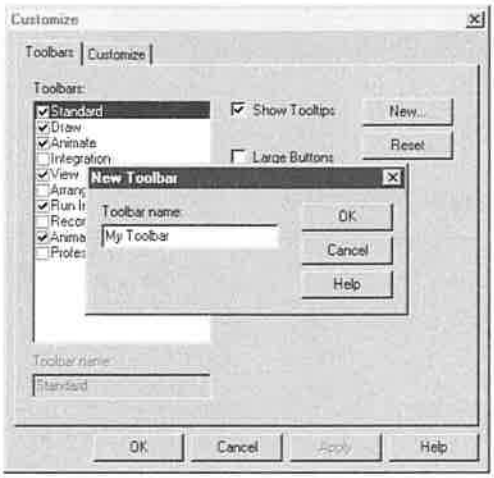
Toolbars Tab

Show Tooltips

Large Buttons

Turn on/off display of individual toolbars

New... Button



By clicking the **New** button, you can create your own toolbar.

Reset Button

The **Reset** button resets the highlighted toolbar to its default configuration.

Customize Tab



The customize tab lets you drag a button to any toolbar to customize it for your use.

To remove a button from a toolbar, hold down the **Alt** key and drag the button off the toolbar.

The Draw Toolbar



Draw Objects

Line, polyline, arc, Bezier curve, box, polygon, ellipse, text

Line Color, Fill Color, and Text Color

Window Background Color

Line Styles and Fill Patterns

Drawing Graphic Objects

1. Select object type from Draw toolbar (line, polyline, arc, Bezier curve, box, polygon, ellipse).
2. Position cross hair at first locator point.
3. Click to establish first point.
4. Move mouse (no button pressed) to position next locator point(s).
5. Click to establish next point(s).
6. Double-click to finish (polyline, polygon, Bezier curve).

Notes

- Shift key constrains mouse movement to create square (box) / circle (ellipse) or to constrain lines to 45-degree increments.
- Grid display and snap (View menu, toolbar)

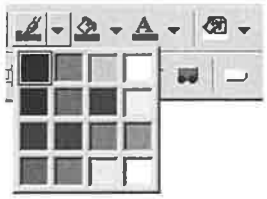
Placing Text

1. From the Draw toolbar, click on the **Text** button. 
2. Type the text in the dialog. Use **Ctrl+Enter** for a new line; **Ctrl+Tab** to insert tabs.
3. From the Text String dialog, click the **Font** button to open a dialog for selecting the text font and style. 

4. Close the dialog(s) by clicking **OK**.
5. Position the cross hair at the bottom left point of the text area and click the left mouse button to place the text.

Changing an Object's Color

1. Click on object(s) to select.
2. Open the color palette corresponding to the characteristic to be changed (i.e., line, fill, text, or background).
3. Click on desired color in color palette.



Notes

- To customize the color palette, right-click on any color cell. Choose the desired color and click **OK** to replace the color in the cell.
- The number of colors that are displayed in the palette may be changed by selecting the **Tools > Options** menu item and clicking on the **Settings** tab.

Selecting and Moving Objects

Selecting Multiple Objects

Box select to enclose all desired objects (drag cursor to define box)

Ctrl+click to create selection set of individual objects

Shift+click to cycle selection among overlapping objects

Select All (Edit menu, Ctrl+A, or use right-click menu) selects all objects in window

Each individual selected object is highlighted

Color, line style, etc., changes apply to all selected objects

Moving Objects

Click and drag selected object to new position. (Click on the edge of boxes, ellipses, and polygons.)

If multiple objects are selected, all are moved.

Copying and Removing Objects: Edit Menu

Clipboard

Cut (Ctrl+X) or right-click menu 

Copy (Ctrl+C) or right-click menu 

Paste (Ctrl+V) or right-click menu 

Duplicate (Ctrl+D or Ins or right-click menu)

Delete (Del)

Undo / Redo

Undo (Ctrl+Z or right-click menu)



Object edit and arrange actions supported (e.g., delete, move, change color)

Unlimited undo

Saving model, editing module clears undo history

Redo (Ctrl+Y or right-click menu)



Restores actions that were previously undone

Toggle Split Screen

Toggles between a split screen containing both a flowchart view and a spreadsheet view simultaneously, or a separate screen for the flowchart view and spreadsheet view.



Arranging Objects: Arrange Toolbar



Adjusting Object Overlap

Bring to front

Send to back

Grouping Objects

Group

Ungroup

Rotating Objects

Vertical flip

Horizontal flip

Rotate 90° — except for OLE objects

Module Alignment and Spacing

Align top

Align bottom

Align left

Align right

Space across

Space down

Changing the View of the Model: View Toolbar



Named Views (also on View menu)

Name and hot key (case sensitive) for views of model

Displayed in the Navigate tree on the Project bar

Moving In / Out

Zoom in (– key)

Zoom out (+ key)

View Controls

View all objects in model (* key)

Return to previous view

Grid, Snap, and Snap-to-Grid Controls

Microsoft® Visual Basic® for Applications

Visual Basic for Applications (VBA) is a component of Arena licensed from Microsoft Corporation. It is the same language technology that powers Microsoft® Office applications, including Microsoft® Word, Excel, PowerPoint®, and Visio® and Arena. It is also the same engine behind Microsoft's own Visual Basic®.

Incorporating VBA into Arena allows for tighter integration by extending the VBA interface to include Arena's own objects, methods, properties, and events. The interface to VBA is a separate window, integrated with Arena, in which you can edit, design, and debug Visual Basic code and forms. With the inclusion of VBA in Arena, Visual Basic developers also will be able to integrate Arena with other programs that support the Microsoft® ActiveX® Automation programming interface.

The Integration toolbar provides an interface to Microsoft Visual Basic for Applications and Arena's Module Data Transfer wizard.



Opens the Module Data Transfer Wizard. This wizard steps you through the process of setting up a method of transferring data from your modules to alternate data sources such as Microsoft Excel, Microsoft Access, and Text.



Opens the Visual Basic Editor (VBE). You can use the VBE to create, view, debug, and run VBA programs stored in an Arena model. The Visual Basic Editor can also be opened from the Tools menu or by pressing **Alt + F11**.



Toggle the VBA Design Mode button between Design and Use mode.

Visio® Link

If you use Visio®, you can immediately put Arena's process-mapping tool to work with the Visio Process Simulator. Created as an add-in to Visio, the Process Simulator includes a Visio template and stencil for mapping processes with shapes designed to mirror the flowchart and data modules in the Basic Process panel. When you've completed your Visio process map using these tools, you're ready to simulate it in Arena!

SMART files SMARTS030 and SMARTS048 will give you good insight in the combined use of Visio and Arena.

Also, Visio diagrams can be converted to Arena models. The link is launched through Arena's **File > Import Visio Drawing** menu. When a drawing is selected, the Visio Link creates a new Arena model file containing modules and draw objects corresponding to the objects in the Visio drawing.

Arena has a shape mapper, which acts as a blueprint for transferring Visio flowchart shapes to the Arena environment. The map can be edited, which allows the mapping of any Visio shape/stencil to any Arena template (including user-created stencils/templates).

Using CAD Drawings

DXF File Import

AutoCAD® release 2002 and all previous versions

Save CAD drawing as DXF file

Arena import options for grouping graphics, clipping at CAD drawing viewport, etc.

Graphics added to model window; edit in Arena

Promote path to use CAD graphic as animation path

Insert as OLE Object

Links or embeds CAD graphics with Arena model

Graphics edited using CAD application

Through object linking, can edit CAD separately from Arena model with automatic updating

Printing

Standard Windows Printers

Print Preview (Toolbar button or File menu)

Print Setup (File menu)

Printer selection

Printer properties (e.g., TrueType font support, graphics quality)

Paper options (e.g., tray, size)

Orientation: landscape or portrait

Printing Model Logic and Animation

*Important Note: Printing is executed **one** model level at a time!*

Print Range Options

Entire Document — Prints the current view and all of the named views (individually in the order they were defined in the model, NOT in the order listed in the Navigate bar). If no named views are defined, only the current view will print.

Specified Pages — Arena uses views to determine page numbers. Specifically, the current view is always Page 1. The rest of the pages are ordered chronologically the way they were entered in the Named Views List. Therefore, the first Named View entered is Page 2, the second Named View added is Page 3, the third Named View defined is Page 4, and so on.

Selection — Prints the portion of the model that has been selected with the cursor. It will scale the selected area/items to a single page.

Printing a Report

Click on the **Print** button () in the Report window to print the active tab.

Exporting a Report

Click on the **Export** button () to export a report to a word processing, database, spreadsheet, HTML, ODBC, or a standard data interchange format. (For example, you can export your report to Microsoft® Excel format and then open it in Excel as a worksheet file.)

Arena Runtime

The Arena runtime feature allows analysts to perform what-if simulation analysis using an Arena model built by someone else. Its capabilities are oriented toward helping analysts change model parameters, perform simulation runs, and analyze the results of these runs.

When opening a model, Arena automatically places a model under runtime restrictions if the model uses an unlicensed feature and the demonstration limits are exceeded. A model must have been saved using a valid commercial Arena license to be opened in runtime mode.

NOTE: Commercial runtime models may be legally loaded and run only by employees of an organization that owns a valid Arena license. Models created using an academic license are not valid for runtime use.

A new Runtime activation is now available that specially configures Arena to load any model regardless of size with runtime restrictions. This feature provides an economical way to extend the benefits of Arena runtimes to users who do not own an Arena license.

When a model enters Runtime mode, an analyst can modify the characteristics of any objects in the model, including module data, object positions, animation pictures, etc., but may not add or delete objects in the model or items in a module repeat group. Full simulation run control support is available.

B: Debugging Tools

Topics covered in this section:

Concepts

- Verifying and validating models
- Debugging models

Arena Constructs

- View > Debug Bar
- View > Runtime Elements Bar
- View > Run Controller

A model can compile without errors, but may produce errors during runtime or provide invalid results when executed. Debugging is the process of identifying and correcting these errors.

The Debug Bar, Runtime Elements Bar, and the Run Controller are tools that Arena provides to aid in the debugging process. These tools enable the user to:

- Interact with the model process logic
- Suspend model execution at critical points
- Examine values of various expression values at varying points of time
- Modify model values

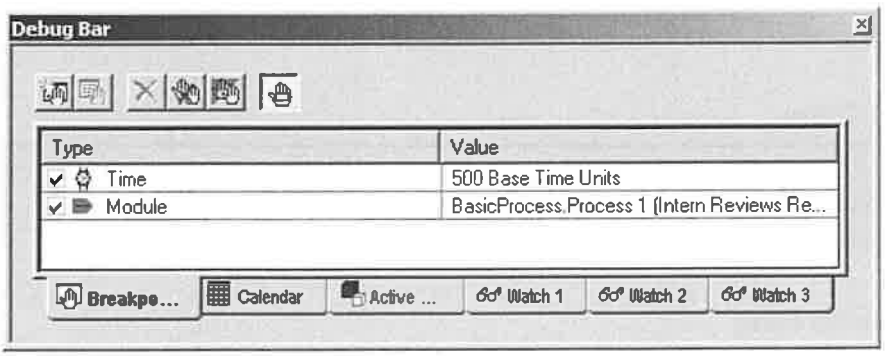
The following sections describe each of the elements within each of the debugging tools.

Concepts and Terminology

The Debug Bar (shown on next page) allows you to monitor the status and control the execution of a running simulation model. The Debug Bar consists of the following windows:

- Watch
- Breakpoint
- Calendar
- Active Entity

To open the Debug Bar, select **View > Debug Bar**.



Watch Window

The Watch Window allows you to monitor the values of expressions within the simulation. This can be beneficial in understanding various expressions as the system changes over time. Common expression types used are:

- Variable Values
- Attribute Values
- Number in Queue
- Resource Availability/Current Utilization

Watch window values are updated whenever the simulation run is interrupted (e.g., paused, stopped on breakpoint, etc.).

Breakpoints

Breakpoints enable the user to pause the simulation at various points in time. Breaks in execution can be established at specific times, when an entity number becomes active, on a module, or based on some model condition. The five types of breakpoints available are:

- Break on Time
- Break on Condition
- Break on Entity
- Break on Module
- Break on Calendar Date Time

Breakpoints are an invaluable aid in the debugging process. They enhance the ability for a modeler systematically to debug and/or verify that model logic is operating as intended.

Calendar

Displays the future events scheduled on Arena's SIMAN event calendar for the running simulation. Event times, entity numbers associated with the events, and event descriptions are displayed in a table format.

Active Entity

The Active Entity window displays the attributes and the current values of the attributes for the active entity. As with other windows, values are updated after the model has been interrupted.

Attribute values can be changed during the model run, by double-clicking within the value field.

NOTE: This is only advisable for model debugging and not for model analysis.

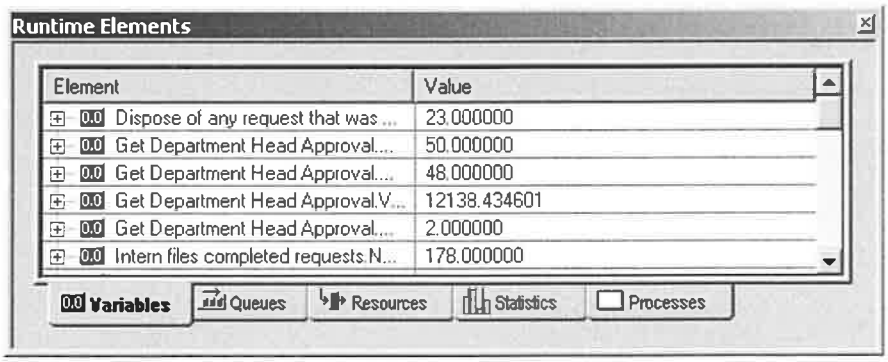
Concepts and Terminology

The Runtime Elements Bar (shown below) allows you to view the properties of several types of model elements during the running simulation. The Runtime Elements Bar consists of the following windows:

- Variables
- Queues
- Resources
- Statistics
- Processes

NOTE: The bar will open after the model has been checked and initialized and closed on model completion.

To initiate the Runtime Elements Bar, select **View > Runtime Elements Bar**.



Often the Runtime Elements Bar is used in conjunction with the Debug Bar and watch windows. One available feature is the ability to copy and paste values from the Runtime Elements Bar into one of the watch windows of the Debug Bar. Another feature is the availability of dragging and dropping values from the Runtime Elements Bar to a watch window (similar to copy/paste).

The object properties in the Runtime Elements Bar are updated whenever the simulation run is interrupted (i.e., paused). Property values that have changed since the object was last viewed are displayed in red.

Variables

The Variables window displays all of the model's default variables as well as any user-specified variables. The value that is displayed is the current value of the variable.

Queues

The Queues window displays all queues within the model and individual queue information, including:

- Number in Queue
- Average Number In Queue, Wait Time, Wait Cost
- Entities in Queue

The Entities in Queue branch enables the user to see not only the position of each entity in the queue, but also all of each entity's attribute values.

Resources

The Resources window displays information for each model resource. Information includes:

- Current State
- Costs (Usage, Idle, Busy)
- Usage
    - Number scheduled, number busy, number seized
    - Average number scheduled, average number busy
    - Utilization

Statistics

The Statistics window displays system and user-defined statistics. Statistics displayed include:

- Tallies
- Time-Persistent
- Counters
- Output
- Frequency

Processes

The Processes window displays information for each Process module with the model. Information includes:

- Number in Process
- Number In
- Number Out
- Costs (VA, NVA, Other, Transfer)
- Times (VA, NVA, Other, Transfer)

Concepts and Terminology

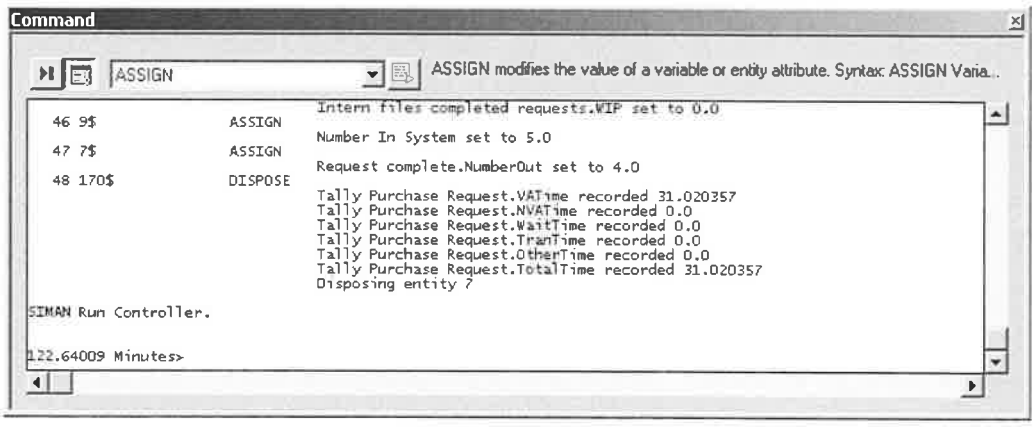
The Run Controller allows you to monitor the status and control the execution of a running simulation model. The Run Controller displays each line of model logic that is taking place during a model run.

The Run Controller (shown below) is invoked by selecting the Command option from the Run menu (**Run > Run Control > Command**). It can be invoked at the start of the simulation run or at any time during execution after the model has been paused. Should the model be in edit mode prior to selecting the Run Controller, the model will check and display the Run Controller at simulation time zero. The Run Controller then prompts you to enter commands interactively from the keyboard.

The command prompt is marked by the current value of simulated time, TNOW. Commands can be executed by typing the command name and any modifying keywords and operands at the prompt, and then pressing Enter.

Some rules to remember:

- Two-or three-letter abbreviations are sufficient to specify any command or keyword when entering a command.
- Unique identification numbers are assigned to each entity. These numbers appear in trace statements as entities move through the model, when using certain commands to view queue contents, and so on.
- Intercepts, traces, and watch points set on entities apply only to the active entity – the entity currently executing blocks.



Some of the more commonly used commands are:

SET TRACE (SE TR)	Displays a trace of all model statements as they are executed. Model execution is suspended at breakpoints.
CANCEL TRACE (CA TR)	Removes trace options. Stops the display of model statements in the Run Controller window.
STEP	Executes the next block or event in the simulation and suspends execution.
SHOW	Displays the value of any expressions involving system- or user-defined variables or the attributes of an entity that has encountered a breakpoint in the model; e.g., show vMyVariable.
VIEW ENTITY	Displays the information about entity EntityID. All attributes of the entity displayed.
VIEW QUEUE	Displays the contents of queues. Produces a listing of the rank and attribute values for entities in the queue.
SET WATCH	Allows you to monitor changes to the value of any expression involving system variables or attributes. Immediately after change to a watched expression occurs, the Run Controller suspends execution of the simulation and prompts for a command.
CANCEL WATCH	Removes watch points on expressions that were previously established with the SET WATCH command.
GO	Begins or resumes simulation execution.
QUIT	Ceases execution of the simulation program without generating a summary report and cancels any remaining replications.

More details on these and other Run Controller commands can be found in Arena Help files.

Debugging Exercise (Debugging Example.doe)

The manager of the purchasing department (Basic Process Workshop) originally had an intern build a model of the purchasing department process. Unfortunately due to time constraints, the model was not completed and fully debugged.

The original workshop model (*Basic Process Workshop.doe*) has had additional model logic and flexibility built into it. The following additions were made to the model:

- An input file has been created to allow the user to conduct various what-if scenarios. Input values changeable in the input sheet are:
    - Percentage of requests less than $2000 for each customer type
    - Percentage of requests over $2000 that gain department head approval for each customer type
    - Customer type percentage
- Requests are now considered to come from two sources, New customers and Long Time customers. An attribute, *aCustomerLevel*, is used to identify the customer types. This attribute is assigned as the requests enter the system. This attribute is randomly selected by using the Discrete distribution with the percentage values of customer type (*vCustomerLevelPercentage*) read from the input file.
- The decision used to determine whether the request is greater than $2000 is based on the type of customer. These values are read from the input file into a variable named *vRequestsLessThan2000Dollars*.
- The decision used to determine whether the department head approves requests greater than $2000 is also based on the type of customer. These values are read from the input file into a variable named *vDeptHeadApproval*.
- Requests not approved by the department head now require the intern to write a rejection document requiring anywhere from 1 minute to a maximum of 6 minutes, but usually requiring 4 minutes.

Open the model, *Debugging Example.doe,* and use the command debugger and other debugging techniques to eliminate the model errors.

Hint 1: There are five known model errors.

Hint 2: Using the default input values, the number of items out is 216 and the intern utilization is 28.55%.

Default values for the inputs are:

Customer Type	% Requests < $2000	% Requests > $2000 Gaining Department Head Approval	Discrete Customer %
New	75	25	20
Long-Time	40	90	100

NOTE: To view the solution, enable the Edit/Run layers (**View > Layers > Enable All**).

C: Commonly Used Variables and Mathematical Expressions

Arena provides many built-in system variables to access and modify data related to a model. This section provides a brief overview of some of the more commonly used variables and mathematical expressions. Manipulation of these elements is simplified with use of the Expression Builder.

Commonly Used Variables

NQ — Number in queue. NQ returns the number of entities in the specified queue.

Example: NQ (Seize Paint Booth.Queue)

MR — Resource capacity. MR returns the number of capacity units currently defined for the specified resource. The Alter module or Schedules option may be used to change the value of MR for a resource. MR is an integer quantity.

Example: MR (Paint Booth)

NR — Number of busy resource units. Each time an entity seizes or preempts capacity units of a resource, the NR variable changes accordingly. NR is not user-assignable; it is an integer value.

Example: NR(Paint Booth)

NREP — Replication number. NREP returns the current replication number, an integer value. It may not be changed by the user.

TFIN — Final simulation time. TFIN is the ending time scheduled for the replication; it is a real-valued quantity. If the ending time is defaulted, TFIN returns a large value (1.0E+20). TFIN may be assigned a value (greater than the current simulation time, TNOW) during a replication. Any subsequent replications use the value of TFIN from the **Run > Setup > Replication Parameters** page.

TNOW — Current simulation time. TNOW records the simulation clock time as the model progresses. After all activities at a particular simulation time have been processed, TNOW is updated to the time of the next activity (e.g., entity event). TNOW is a real-valued quantity; it is not user-assignable.

J — Search index variable. J is an integer, user-assignable variable. It is used in the Search and FindJ modules to return the selected index value, based on a search criterion and range. J may be used within a model for other purposes as well; however, execution of a Search or FindJ module will give J a new value.

Mathematical Expressions

Mathematical expressions may be formed using combinations of integers, constants, attributes, variables, or random distributions. Conditional statements may be formed and are evaluated as numerical expressions; a true condition is assigned a value of 1, and a false condition is assigned a value of 0. Numerical expressions themselves are also evaluated on a True/False basis when entered for a Condition operand; 0 is False and any non-zero value is True.

The following operators are available, and expressions are evaluated using the operator priorities listed. Innermost parentheses are evaluated first according to these priorities. Operators with the same priority are evaluated from left to right.

Operator		***Operation***	***Priority (1 = Highest)***
**		Exponentiation	1
/		Division	2
*		Multiplication	2
-		Subtraction	3
+		Addition	3
.EQ.	==	Equality comparison	4
.NE.	<>	Non-equality comparison	4
.LT.	<	Less than comparison	4
.GT.	>	Greater than comparison	4
.LE.	<=	Less than or equal to comparison	4
.GE.	>=	Greater than or equal to comparison	4
.AND.	&&	Conjunction (and)	5
.OR.	||	Inclusive disjunction (or)	5

NOTE: == is a logical operator and = is an assignment operator. Use == to test whether two items have the same value; use = to set a value, as in the Assign module.

Math Functions

The following 20 standard mathematical functions are also available. Each function takes a parameter list enclosed in parentheses. These parameters may be specified as constants or expressions when used in a model. The geometric functions (ACOS, ASIN, ATAN, HCOS, HSIN, HTAN, COS, SIN, TAN) take a value specified in radians.

Function	*Description*
ABS(a)	Absolute value
ACOS(a)	Arc cosine
AINT(a)	Truncate
AMOD(a1 , a2)	Real remainder, returns (a1-(AINT(a1/a2)*a2))
ANINT(a)	Round to nearest integer
ASIN(a)	Arc sine
ATAN(a)	Arc tangent
COS(a)	Cosine
EP(a)	Exponential (ea)
HCOS(a)	Hyperbolic cosine
HSIN(a)	Hyperbolic sine
HTAN(a)	Hyperbolic tangent
MN(a1 , a2 , ...)	Minimum value
MOD(a1 , a2)	Integer remainder, same as AMOD except the arguments are truncated to integer values first
MX(a1 , a2 , ...)	Maximum value
LN(a)	Natural logarithm
LOG(a)	Common logarithm
SIN(a)	Sine
SQRT(a)	Square root
TAN(a)	Tangent

D: Expression Builder

The Expression Builder is a tool to assist in the creation of mathematical, logical, and Arena-specific expressions.

Examples:

Mathematical: Counter + 1 — Incrementing a user-defined variable.

Logical: (Counter >= 1)AND(Counter <= 4) — Checking to see if the variable is greater than or equal to 1 and less than or equal to 4.

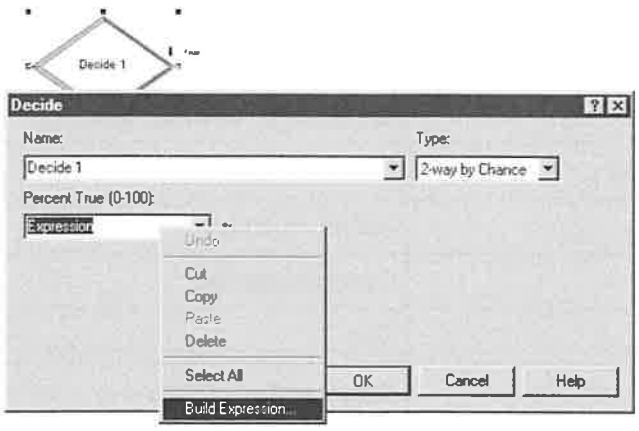
Arena: NQ(Machining Process.Queue) — The current number in the queue for the Machining Process.

These expressions can be easily created and inserted where needed, regardless of user knowledge of the format or syntax used by Arena.

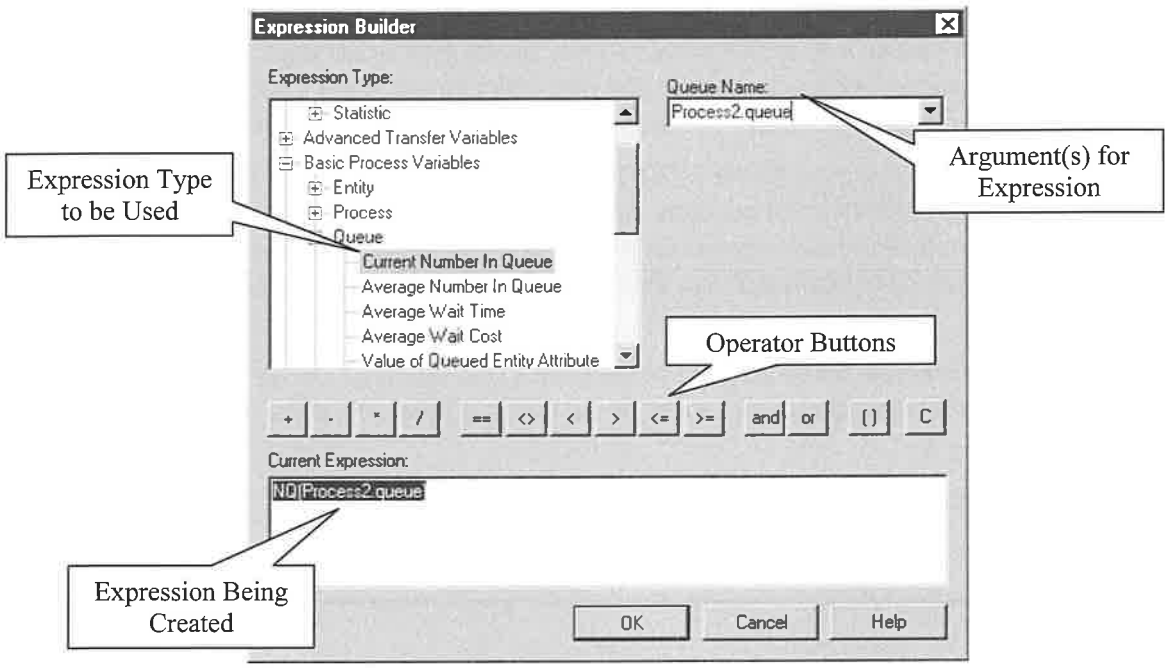
How To Use

The Expression Builder can be selected from any module or animation object (e.g., Variable, Plot, Histogram) dialog field where you may enter an expression.

To start the Expression Builder, click on the field or cell where you want to enter the expression, then right-click the mouse. Select **Build Expression** from the right-click menu.



The Expression Builder is composed of four main sections.



The *Expression Type* allows for quick selection of the elements of the expression to be used. When selected, the element is automatically pasted into the *Current Expression*.

Arguments for the expression are displayed in the upper right-hand portion of the Expression Builder. These arguments are unique to the expression type being used.

The *Operator Buttons* are commonly used mathematical and logical operators. The "C" is a quick method of clearing the currently displayed expression.

The *Current Expression* box is the expression that is being created. Expressions can be created with the aforementioned portions of the Expression Builder or in combination with directly typing in the desired expression.

By clicking **OK**, the current expression will be copied and pasted to the location where the Expression Builder was generated.

E: Costing

When creating a simulation model, you may be interested in keeping track of the cost of processing entities or the cost of using different resources. Arena's built-in costing functions give you the ability to assign different costs based on activities in the model.

Costs can be accrued for each entity based upon the amount of time that an entity uses a resource, as well as per use costs. As an entity travels through the system, costs are accumulated and then reported for the entities. Average, minimum, and maximum cost per each entity type is recorded. Entity costs are also divided into processing and waiting costs.

Resource costs can be assigned and reported for any resources that are included in your model. These costs will be accrued for the following three categories: idle, busy, and per use.

Costing is optional when creating a model of your process.

Value- and Non-Value-Added Activities

An important function of Arena's costing function is for the organization's activities to be defined as value-added or non-value added.

Value-added activities are those for which the customers are usually willing to pay (in some way) for the service. Value-added is determined if the output of the activity is directly related to customer requirements, service, or product, as opposed to an administrative or logistical outcome that services the providing organization. For instance, if the output of an activity were an inventory report or update for products (for which there are customers), the output would be non-value added, but necessary to the organization; i.e., "overhead."

Non-value-added activities are those that create waste, result in delay of some sort, add costs to the product/s, or for which the customer is not willing to pay.

How do I add costing to my model? Where do I input costs?

Entity Module

Holding Cost/Hour – Hourly cost of processing the entity through the system. This cost is incurred when the entity is anywhere in the system, either in queue waiting, or in value- or non-value added processes.

Initial VA Cost – Initial cost value that will be assigned to the value-added cost attribute of the entity. The attribute will be increased when an entity goes through a value-added process step.

Initial NVA Cost – Initial cost value that will be assigned to the non-value-added cost attribute of the entity. The attribute will be increased when an entity goes through a non-value-added process step.

Initial Wait Cost – Initial cost value that will be assigned to the waiting cost attribute of the entity. This attribute will be increased when an entity is waiting in a queue, including waiting to be batched or waiting for resource(s) at a Process module.

Resource Module

Busy/Hour – Cost per hour of a resource that is processing an entity. The resource becomes busy when it is originally allocated to an entity and becomes idle when it is released. During the time when it is busy, cost will accumulate based on the busy/hour cost.

Idle/Hour – Cost per hour of a resource that is idle. The resource is idle while it is not processing an entity. During the time when it is idle, cost will accumulate based on the idle/hour cost.

Per Use – Cost of a resource on a usage basis, regardless of the time for which it is used. Each time the resource is allocated to an entity, it will incur a per-use cost.

Process Module(s)

Allocation – Determines how the processing time will be allocated to the entity and process costs. The process may be considered to be value-added or non-value-added, and the associated cost will be added to the appropriate category for the entity and process.

What costs are calculated?

There are three categories of times and costs that are accumulated for each entity in the system, as well as each process through which an entity moves:

Wait Time – The period of time from when the entity enters a queue (in either a process module to await a resource(s), or in a batch module to await batching) until the entity exits the queue (seizes the resource(s) or is batched).

Wait Cost – (Holding cost of the entity * Wait Time).

Value-Added Time – Is accumulated when an entity incurs a delay at a value-added process.

Value-Added Cost – (Holding cost of the entity * Value Added Time) + (Resource cost for the entity * Value-Added Time) + Resource Usage Cost.

Non-Value-Added Time – Is accumulated when an entity incurs a delay at a non-value-added process.

Non-Value-Added Cost – (Holding cost of the entity * Non-Value-Added Time) + (Resource cost for the entity * Non-Value-Added Time) + Resource Usage Cost.

Additionally, **Total Cost** and **Total Cycle Time** are calculated.

Where are the calculated costs in my reports?

Some form of cost statistic is provided in all reports generated by the simulation.

Reports – Summary of Processes

Gives process time, count, and cost information per replication for all processes in a model. This allows you to compare easily throughput, cost, and time information by processes.

Contains the categories:

Process Throughput

- Number In
- Number Out
- Total

Process Costs

- Total Value-Added Cost
- Total Non-Value-Added Cost
- Total Wait Cost
- Total Cost

Process Times

- Total Value-Added Time
- Total Non-Value-Added Time
- Total Wait Time
- Total Time

Reports – Summary of Resources

Gives resource cost and usage information per replication for all resources in a model.

Contains the categories:

Resource Costs

- Busy Cost
- Idle Cost
- Usage Cost
- Total

Number Busy

- Average
- Maximum

Utilization

- Average
- Maximum

Reports – Summary of Entities

Gives entity summary information per replication for all entity types in the model.

Contains the categories:

Entity Throughput

- Number In
- Number Out

Average Costs

- NVA Cost
- VA Cost
- Wait Cost
- Total

Average Time

- NVA Time
- VA Time
- Wait Time
- Total

Auto Loan Process – Costing Embellishment (Auto Loan - Costing.doe)

Management has asked us to determine the cost associated with the four major activities of the loan authorization process (Auto Loan - Hierarchy.doe). Accounting has provided us with the following information:

Staff Name	***$/Hour (Idle and Busy)***
Loan Officer	$ 10.00
Evaluation Clerk	$ 6.50
Credit Clerk	$ 6.50

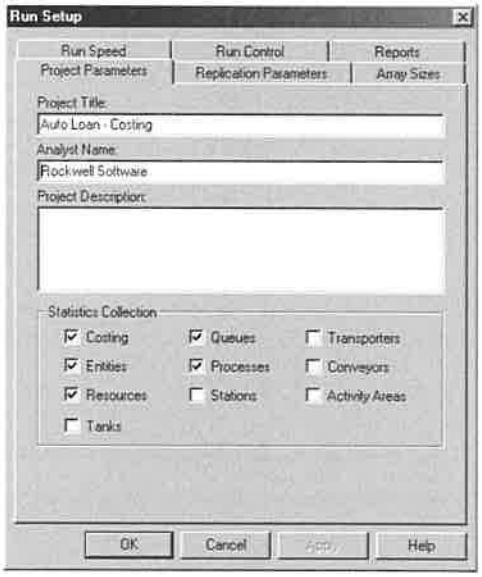
Activity	***Allocation***
Review Application	Non-Value Added
Evaluation and Verification	Value Added
Credit Check	Value Added
Create Acceptance/Rejection Document	Value Added

Resource Module – Loan Officer

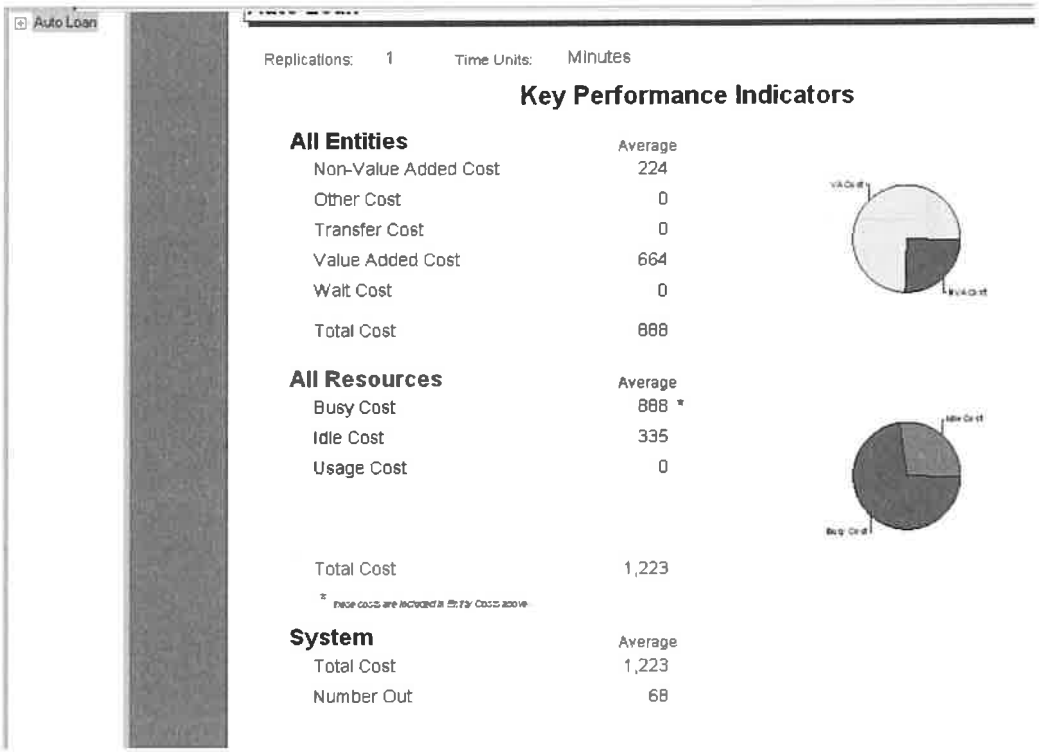


Enabling the Cost Report

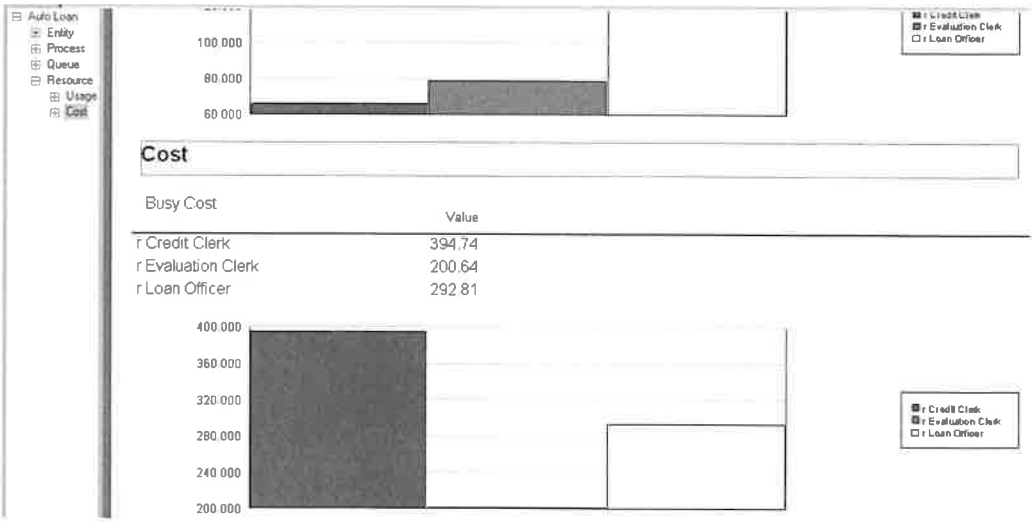
To enable the cost report, check the costing option in **Run > Setup** dialog.



Auto Loan Part 2 Model Costing Embellishment — Summary of Processes



Auto Loan Part 2 Model Costing Embellishment — Summary of Resources



Auto Loan Application Model Costing Embellishment — Summary of Entities

Entity

Cost

Wait Cost	Average	Half Width	Minimum Value	Maximum Value
Complete Applications	0.00	(Insufficient)	0.00	0.00
Incomplete Applications	0.00	(Insufficient)	0.00	0.00

Other Cost	Average	Half Width	Minimum Value	Maximum Value
Complete Applications	0.00	(Insufficient)	0.00	0.00
Incomplete Applications	0.00	(Insufficient)	0.00	0.00

Transfer Cost	Average	Half Width	Minimum Value	Maximum Value
Complete Applications	0.00	(Insufficient)	0.00	0.00
Incomplete Applications	0.00	(Insufficient)	0.00	0.00

Total Cost	Average	Half Width	Minimum Value	Maximum Value
Complete Applications	13.2425	(Insufficient)	6.7118	15.8946
Incomplete Applications	2.5374	(Insufficient)	2.2423	2.8582

Costing Workshop (Costing Workshop.doe)

Management has asked you to determine the cost associated with the production of two new products, the Wizzigig and the Walligog. The file *2NewProducts.doe* contains a completed model of the production process for these two items. The model does not include any costing information. Using the following data furnished by accounting, embellish the model provided and determine the information requested by management.

Resource Name	***Idle Cost/Hour***	***Busy Cost/Hour***
Extruding Machine	$15.00	$68.00
Walligog Finisher	$10.00	$44.00
Wizzigig Finisher	$10.00	$47.00
Painter	$10.50	$11.50
All Inspectors	$9.50	$9.50

Activity	***Allocation***
Extruding	Value Added
Finishing	Value Added
Painting	Value Added
Inspecting	Non-Value Added

F: Additional Features and Methodologies

Concepts and Terminology – Using Conveyors

Conveyors

Physical devices that move entities from one station to another.

Entities *access* conveyors to gain control of space on the conveyor.

Entities are *conveyed* from one station to another.

Entities *exit* conveyors when they are removed from the conveyor.

The conveyor must be defined using the Conveyor spreadsheet.

The travel distance(s) must be defined using the Segment spreadsheet.

Each conveyor is made up of a finite number of equal-length cells.

Conveyor Operations

An entity attempting to access a conveyor is held in a queue until the defined amount of unoccupied space is available at the point of entry.

The point of entry is defined as a station associated with an end of a segment of the conveyor.

The space is defined in terms of the number of cells the entity requires.

The unoccupied cells must be adjacent.

Once on the conveyor, the entity does not move until it is conveyed to its destination station.

When the entity arrives at the destination station, it must be exited or removed from the conveyor.

Conveyors can be straight line or loops, but a single conveyor cannot branch into two lines; different conveyors must be used to model this last situation.

Non-accumulating Conveyors

Equivalent to a standard belt or bucket conveyor where the spacing between entities on the conveyor does not change.

All entities on a conveyor move at the same speed.

An entity that has accessed the conveyor, but not conveyed, stops the entire conveyor from moving.

An entity that has arrived at its exit point, but has not been exited or removed, stops the entire conveyor from moving.

If the time delay between the access and convey or the arrival of an entity at its exit point and the exit is zero, the effect is that the conveyor does not stop.

Accumulating Conveyors

Equivalent to a standard accumulating roller conveyor where, if an entity is stopped on the conveyor, the following entities continue to move until they arrive at the stopped entity.

The spacing or number of cells an entity requires to access the conveyor can be different from the spacing when entities accumulate.

All entities on a conveyor move at the same speed (except for any stopped entities).

Accumulating conveyors are often used to model power-and-free transfer systems.

If no accumulation occurs, both types of conveyors operate in the identical manner, although non-accumulating require less simulation execution time.

Conveyors (Advanced Transfer panel)

The Conveyor spreadsheet defines the operating characteristics of a conveyor

Standard characteristics are cell size, velocity, type, and a referenced segment set

Segments (Advanced Transfer panel)

Defines the conveyor length as well as the starting and ending stations

Conveyor segments are unidirectional

Each segment references a segment set to which it belongs

Additional Advanced Transfer modules

Enter: Incorporates a Station, unload delay, and removal from incoming material handling (Release Resource, Free Transporter, or Exit Conveyor)

Leave: Incorporates loading onto material handling (Seize Resource, Request Transporter, Access Conveyor), load delay, and transfer (Route, Transport, or Convey) to destination station

Toy Paint Shop with Conveyors (Appendix F_Conveyors.doe)

The manager responsible for the Toy Paint Shop would like to incorporate conveyance systems to reduce the movement of parts manually. The manager would like to replace the static buffer area prior to paint with an accumulating conveyor after the prepping area. Non-accumulating conveyors also are to be placed after the painting operation.

The accumulating conveyor will have the same spatial constraints as the static buffer area. Only three toys can be placed on the conveyor at a time. Each toy is 10 ft in length with the conveyor being 30 ft long. The conveyor operates at a rate of 20 ft/min. Toys need to be removed from the conveyor prior to painting.

After the painting operation, toys are placed on one of two non-accumulating conveyors. Conveyors are separated based on part type to make later batching easier. Both of these conveyors are non-accumulating, 50 ft in length, operating at 10 ft/min. These conveyors flow from the paint station to the inspection station. Toys are removed from the conveyor after reaching the inspection station. (No load or unload time is included in the model.)

The painting operation has also changed. Due to a need for a more detailed paint job, painting now requires 10 minutes per part.

Run the model for five 8-hour days. The base time units should be minutes.

Examine the effect of modeling the conveyors explicitly on cycle time and toy throughput. Also examine conveyor statistics.

Modeling Approach

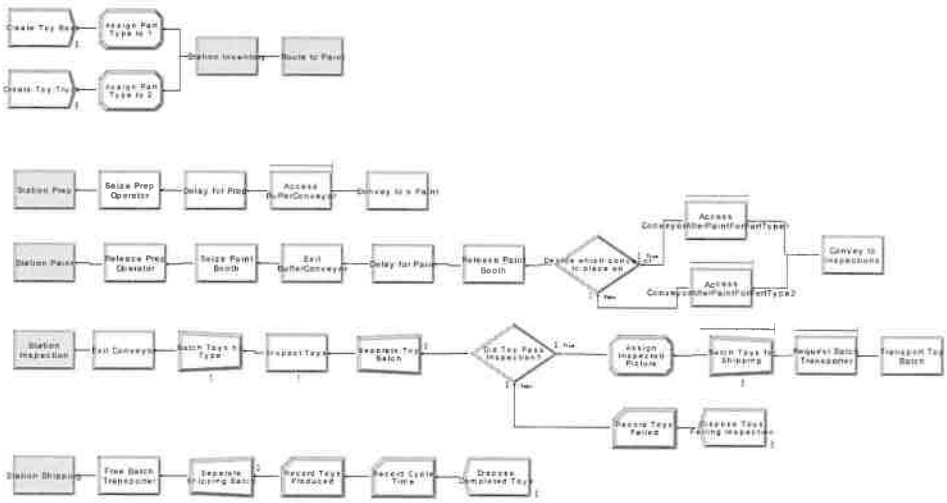
Base Model: Use the model from the section: Advanced Modeling Techniques Exercise 1 (*Advanced Process Modeling Exercise 8-1.doe*) as the base.

Conveyors: Use the Conveyor and Segment spreadsheets to define the conveyors.

Prep Station: Replace the Route with Access and Convey.

Paint Station: Exit the conveyor before processing. Require 10 minutes for processing. After processing, Access one of two conveyors based on part type and Convey to the inspection station.

Inspection Station: Exit the conveyors before processing and batching.



Define Conveyor Parameters — Use the Conveyor spreadsheet

Conveyor - Advanced Transfer

	Name	Segment Name	Type	Velocity	Units	Cell Size	Max Cells Occupied	Accumulation Length	Initial Status	Report Statistics
1	BufferConveyor	BufferConveyor.Segment	Accumulating	20	Per Minute	1	10	10	Active	☑
2	ConveyorAfterPaintForPartType	ConveyorAfterPaintForPartType1.Segment	Non-Accumulating	10	Per Minute	1	10	1	Active	☑
3	ConveyorAfterPaintForPartType	ConveyorAfterPaintForPartType2.Segment	Non-Accumulating	10	Per Minute	1	10	1	Active	☑

Double-click here to add a new row.

Name	BufferConveyor
Segment Name	BufferConveyor.Segment
Type	Accumulating
Velocity	20
Units	Per Minute
Cell Size	1
Max Cells Occupied	10
Accumulation Length	10

Name	ConveyorAfterPaintForPartType1
Segment Name	ConveyorAfterPaintForPartType1.Segment
Type	Non-Accumulating
Velocity	10
Units	Per Minute
Cell Size	1
Max Cells Occupied	10

Repeat for Conveyor for Part Type 2.

Remember: Velocity is a unit of measure per base time unit. Be consistent with your units of measure. If your distance is measured in feet, your velocity should be based on feet.

Example: Base Time Units are Minutes

Distance Units are Feet

Velocity must be Feet per Minute

Define Conveyor Segments — Use the Segment spreadsheet

Segment - Advanced Transfer

	Name	Beginning Station	Next Stations
1	BufferConveyor.Segment	s Prep	1 rows
2	ConveyorAfterPaintForPartType1.Segment	s Paint	1 rows
3	ConveyorAfterPaintForPartType2.Segment	s Paint	1 rows

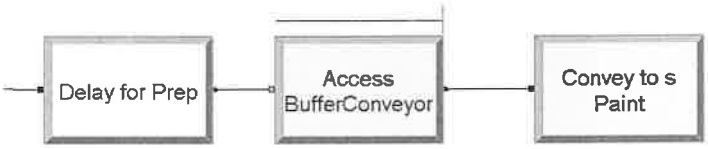
Double-click here to add a new row.

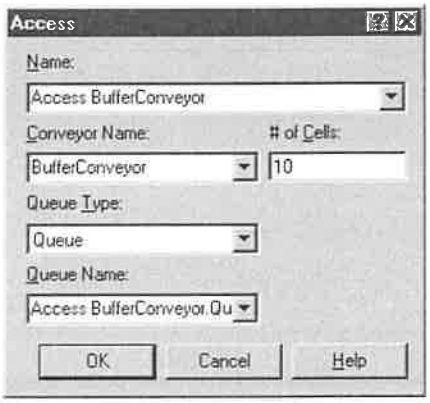
Name	BufferConveyor.Segment
Beginning Station	s Prep
Next Stations	1 row
(s Prep to) s Paint	30

Name	ConveyorAfterPaintForPartType1.Segment
Beginning Station	s Paint
Next Stations	1 row
(s Paint to) s Inspection	50

Name	ConveyorAfterPaintForPartType2.Segment
Beginning Station	s Paint
Next Stations	1 row
(s Paint to) s Inspection	50

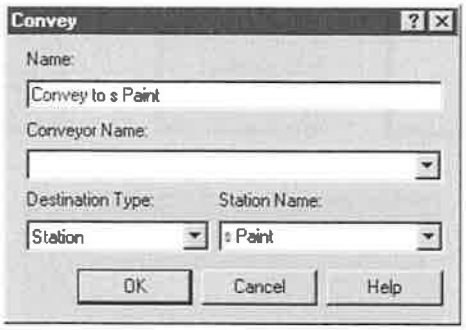
Access Conveyor — Use the Access module





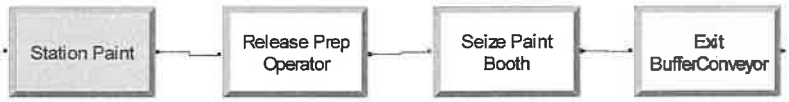
Name	Access BufferConveyor
Conveyor Name	BufferConveyor
# of Cells	10
Queue Type	Queue
Queue Name	Access BufferConveyor.Queue

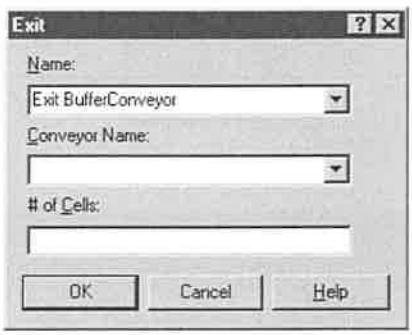
Move the Entity on Conveyor — Use the Convey module



Name	Convey to s Paint
Conveyor Name	
Destination Type	Station
Station Name	s Paint

Exit the Conveyor after Seizing Paint Booth — Use the Exit module

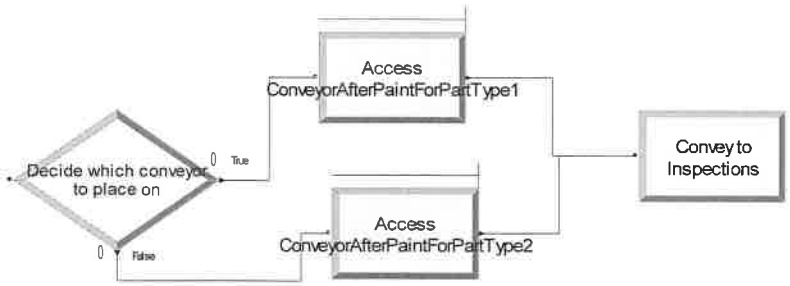


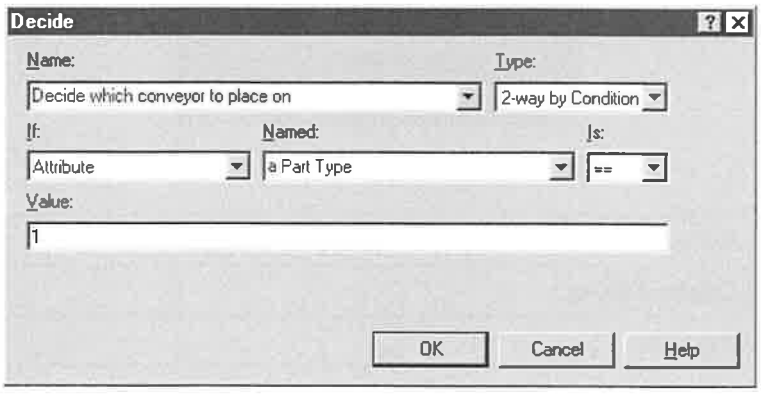


Name	Exit BufferConveyor
Conveyor Name	
# of Cells	

NOTE: By default, Arena will remove the entity from all cells it occupies of the last conveyor it accessed.

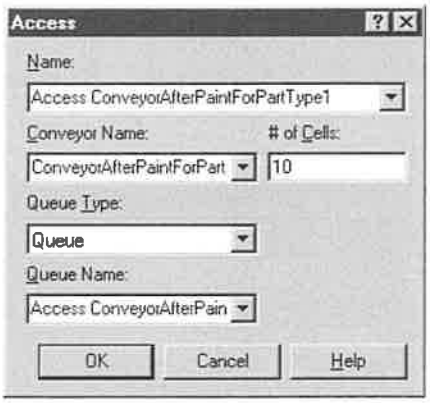
Decide Which Conveyor to Access — Use the Decide module





Name	Decide which conveyor to place on
Type	2-way by Condition
If	Attribute
Named	a Part Type
Is	==
Value	1

Access Conveyor for Part Type 1 and Part Type 2



Access - ConveyorAfterPaintForPartType1

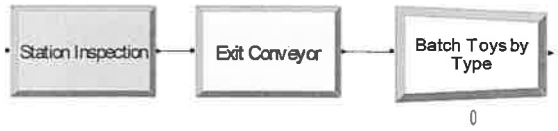
Name	Access ConveyorAfterPaintForPartType1
Conveyor Name	ConveyorAfterPaintForPartType1
# of Cells	10
Queue Type	Queue
Queue Name	Access ConveyorAfterPaintForPartType1.Queue

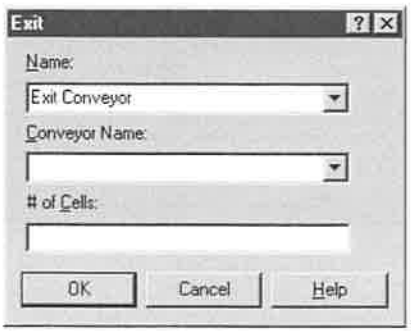
Repeat for Access ConveyorAfterPaintForPartType2

Convey

Name	Convey to Inspections
Conveyor Name	
Destination Type	Station
Station Name	s Inspection

Exit the Conveyors at Inspection — Use the Exit module

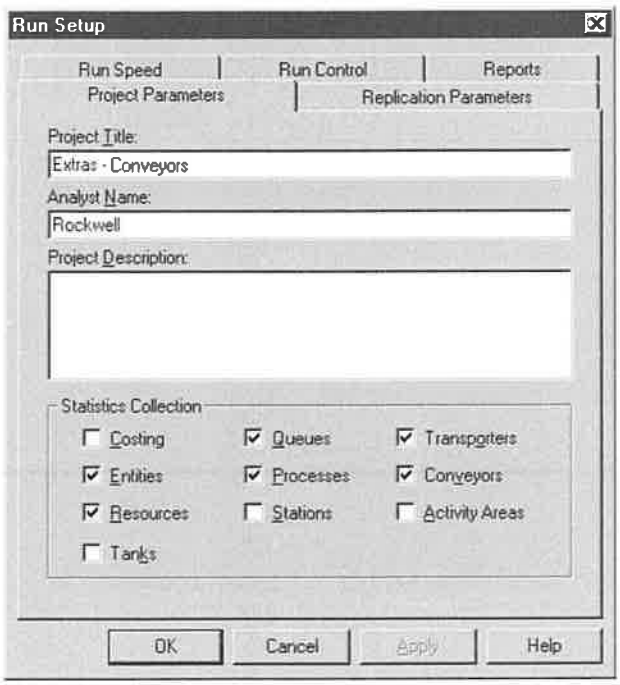




Name	Exit Conveyor
Conveyor Name	
# of Cells	

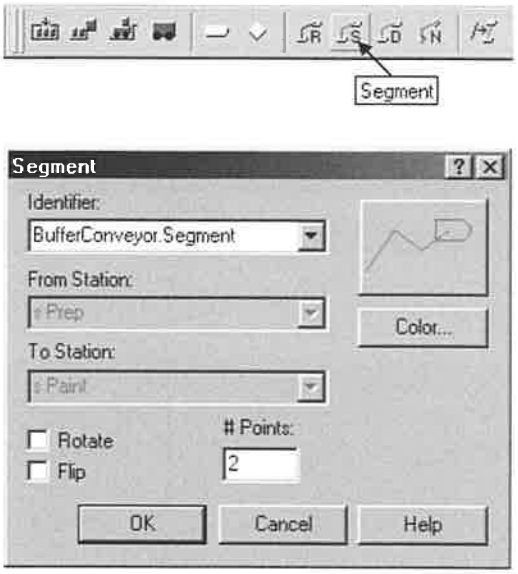
Collecting Conveyor Statistics

In the **Run > Setup** dialog (Project Parameters tab), click on **Conveyors** to turn on collection of conveyor statistics.



Animation

Replace the Route paths with Segment paths among the arrival stations, cutting stations, and trim.



Concepts and Terminology – Using Store

Store

Storages are useful for displaying entity animation while an entity undergoes processing in other modules. Additionally, statistics may be kept on the number of entities in storage.

Unstore

The Unstore module removes an entity from storage. When an entity arrives at the Unstore module, the storage specified is decreased and the entity immediately moves to the next module in the model.

Advanced Set

The Advanced Set is similar to the Set module. This module permits set for queue sets, storage sets and other sets (including variables and expressions) and their respective members. A set defines a group of similar elements that may be referenced via a common name and a set index. The elements that make up the set are referred to as the members of the set.

Toy Paint Shop with Conveyors (Appendix F_Store.doe)

The Toy Paint Shop manager would like to present the model to high-level management. One thing that she would like see in the model is individual areas for the different part types to sit while they wait to be batched for inspection. She wishes to view these prior to batching and then remove them from view after batching has taken place.

The batch process has also changed. Due to machine improvements, items are now batched in threes prior to inspection. Also, because of the change in inspection batching, eliminate the queue priority prior to prep.

Run the model using the previous conditions and view the animation to ensure that the manager request is being met.

Examine the effect of modeling the batch size change on cycle time and toy throughput.

Modeling Approach

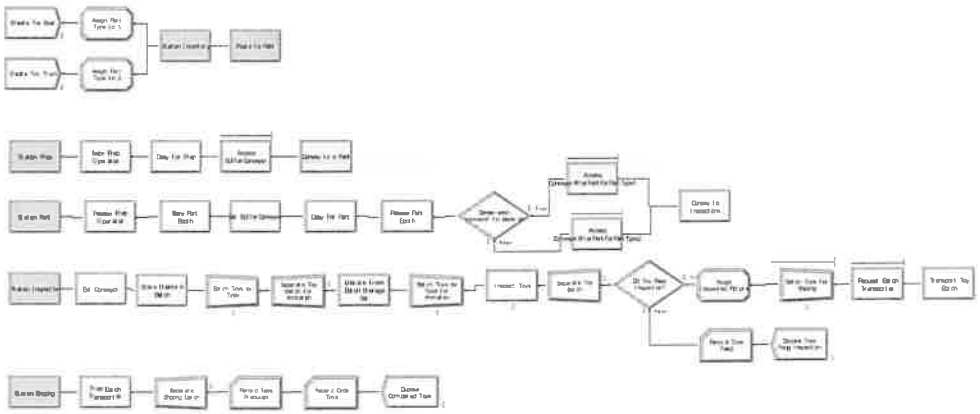
Base Model: Use the model from the section: Toy Paint Shop with Conveyors (Appendix F_Conveyors.doe) as the base.

Storages: Use the Storage spreadsheets to define the storages.

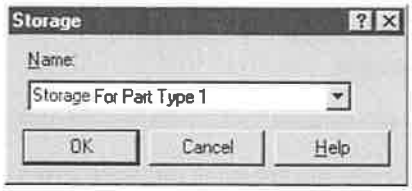
Advanced Set: Create a storage set using the Advanced Set spreadsheets.

Paint Station: Exit the conveyor before processing. Require 10 minutes for processing. After processing, Access one of two conveyors based on part type and Convey to the inspection station.

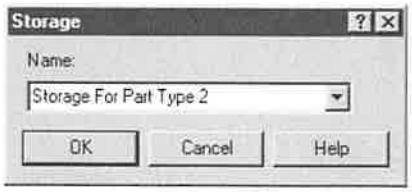
Inspection Station: Exit the conveyors before processing and batching.



Create Storages — Use the Storage spreadsheet module

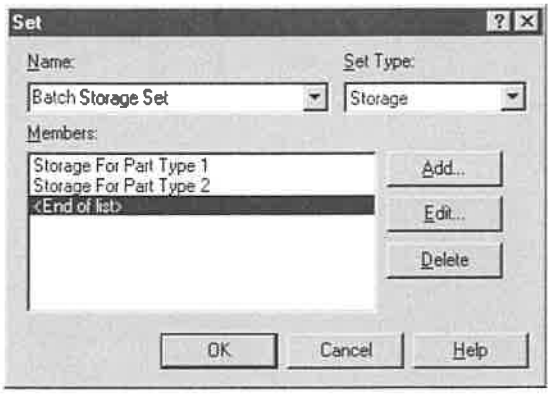


Name	Storage For Part Type 1



Name	Storage For Part Type 2

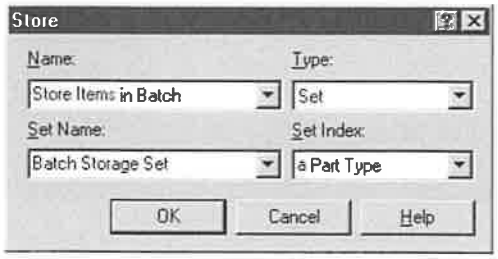
Define Advanced Set — Use the Advanced Set spreadsheet



Name	Batch Storage Set
Set Type	Storage
Members	
1	Storage for Part Type 1
2	Storage for Part Type 2

Store Parts — Use the Store module

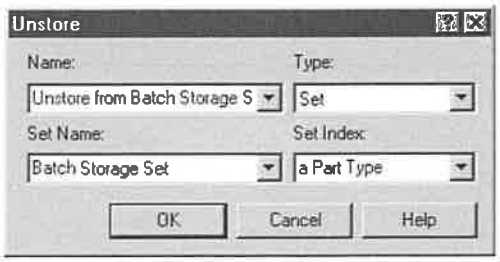




Name	Store Items in Batch
Type	Set
Set Name	Batch Storage Set
Set Index	a Part Type

Unstore Parts — Use the Unstore module





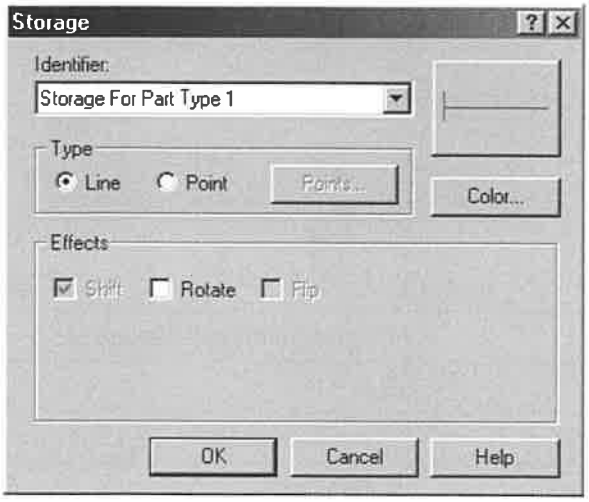
Name	Unstore from Batch Storage Set
Type	Set
Set Name	Batch Storage Set
Set Index	a Part Type

Note: Items need to be separated prior to unstoring and rebatched prior to inspection.

Animation

Replace the Queue in front of Inspection with Storages for each Part Type.





G: Arena 3D (covered in separate course)

H: Statistical Distributions

General Terminology

Minimum — the smallest value

Maximum — the largest value

Mean — the average value

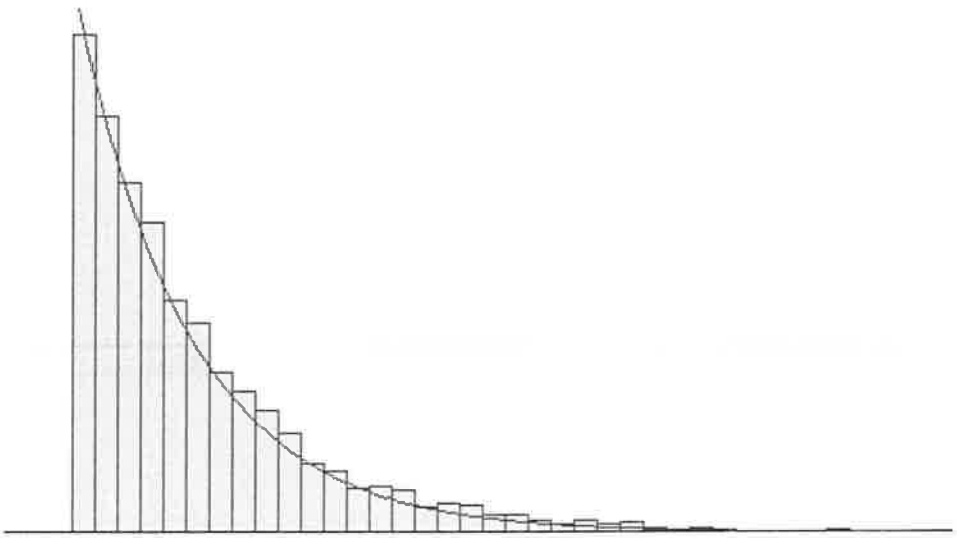
Std Dev — the standard deviation about the mean

Mode — the most likely or common value

Common Statistical Distributions

Exponential Distribution

EXPONENTIAL(Mean) or
EXPO(Mean)



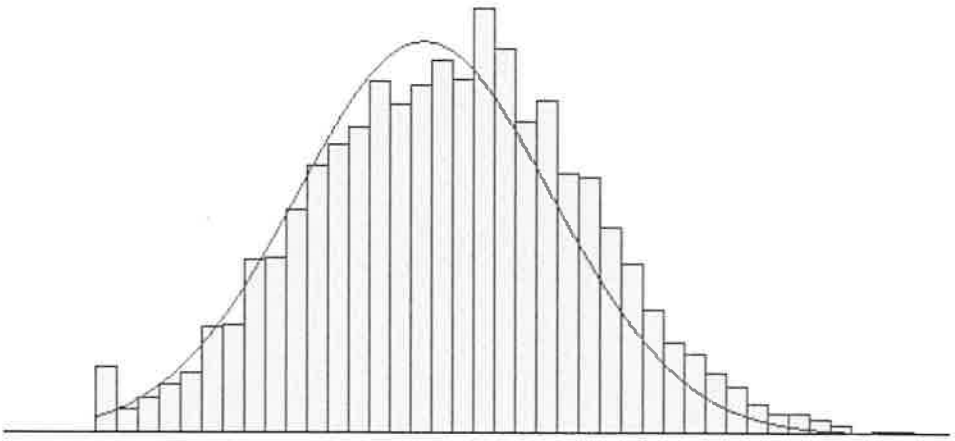
Typical uses are:

Arrivals to the system

Time between random failures

Normal Distribution

NORMAL(Mean, StdDev) or
NORM(Mean, StdDev)

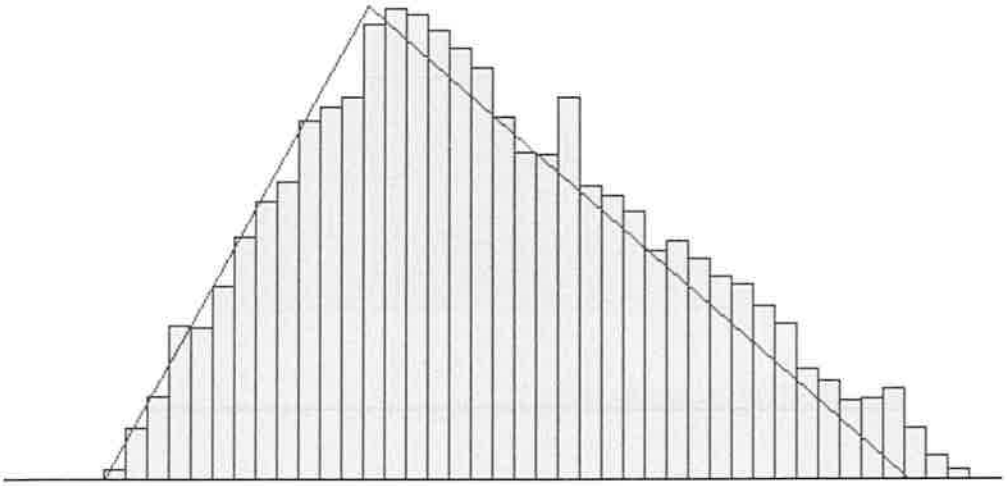


Typical uses are:

Process / repair times

Triangular Distribution

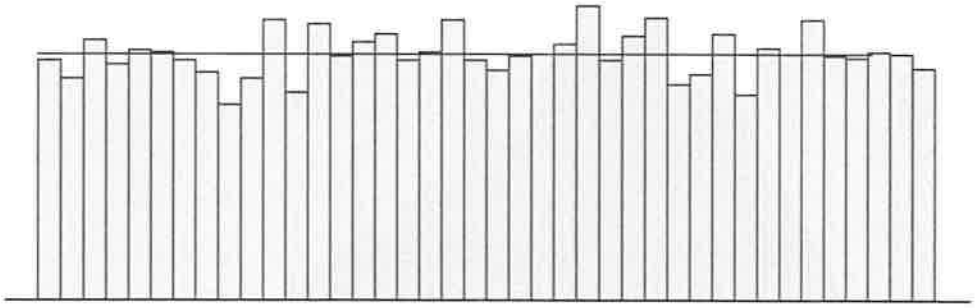
TRIANGULAR(Min, Mode, Max) or
TRIA(Min, Mode, Max)



Approximates many processes and may take on shape of many distributions

Uniform Distribution

UNIFORM(Min, Max) or
UNIF(Min, Max)



All permissible values are equally likely

Additional Distributions

ERLANG(Mean, k)

Mean = exponential mean

k = Erlang parameter

GAMMA(Beta, Alpha)

Beta = scale parameter

Alpha = shape parameter

JOHNSON(Gamma, Delta, Lambda, Xi)

Gamma = shape parameter 1

Delta = shape parameter 2, > 0

Lambda = scale parameter, > 0

Xi = location parameter

LOGNORMAL(Mean, StdDev)

Mean = lognormal mean

StdDev = lognormal standard deviation

POISSON(Mean)

Mean = distribution mean

WEIBULL(Beta, Alpha)

Beta = scale parameter

Alpha = shape parameter

CONTINUOUS(P_1,V_1,P_2,V_2,...)

P_k = cumulative probabilities

V_k = values of dependent random variable

Value returned is computed using linear interpolation within the range of two successive cumulative probability values

DISCRETE(P_1,V_1,P_2,V_2,...)

P_k = cumulative probabilities

V_k = values of dependent random variable

Value returned based on cumulative probabilities specified

NOTE: All distributions may be specified with an additional parameter representing the random stream number

I: Input Data & Input Analyzer

Rockwell Automation

Types of Input Data

When developing a simulation model, the analyst will be concerned with both input and output data. The input data is discussed in this section. The output data or performance results is discussed in the Statistical Analysis section.
There are three types of input data that are required for any simulation model:

– Arrival Patterns
– Decision Probabilities
– Process Times / Move Times / Failure Times

Types of Input Data

A model will be driven by some type of arrival of entities to the system. The arrival pattern may be constant or random. If constant, it may be a given constant time between arrivals or it might be a schedule of entities (not to be confused with the Arrivals Schedule which will be discussed later) arriving to the system. If the arrival is random, the rate may be some type of distribution or combination of distributions.

As the modeler, you must determine how entities are arriving to your system.

Types of Input Data

Once entities are in the model and are using resources, there will be delay times associated with the use of these resources and the movement of entities from one location to another.

These explicit time delays can be constants, random distributions, or an expression which can be a combination of various mathematical expressions.

As entities move through the model there may be points at which a probabilistic decision occurs. The values of these probabilities are another type of data that must be collected.

4

Types of Input Data

Each of these types of data can be expressed in many ways dependent upon the system characteristics. Arena allows the user the ultimate flexibility of describing any of the types of data to whatever level of detail is required. In the following sections the types of distributions available and the methods of fitting data to those distributions will be discussed.

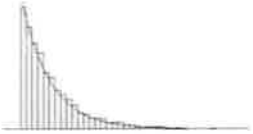
The pitfalls will also be discussed along with how to avoid or at least minimize the impact of data collection, which is the most dreaded part of a simulation project.

5

Rockwell Automation

Common Statistical Distributions

Exponential Distribution
EXPONENTIAL(Mean) or EXPO(Mean)

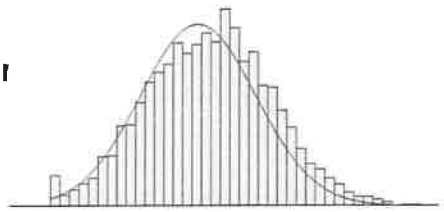


Typical uses are arrivals to the system, time between random failures, and service times that are highly variable. This distribution is quite useful not only because it mimics time between arrivals so well, but because it only requires a mean parameter which may be readily available. In addition, the exponential distribution can be used to determine how sensitive the model is to a given piece of data. This is accomplished by first running the model with the data as the average of the delay and then running the model with the data as the mean of the exponential. This allows the model to experience the most variability for the given data. And of course the exponential would be used if the raw data so indicated. Note that the exponential is not bound

6

Common Statistical Distributions

Normal Distribution
NORMAL(Mean, StdDev) or
NORM(Mean, StdDev)

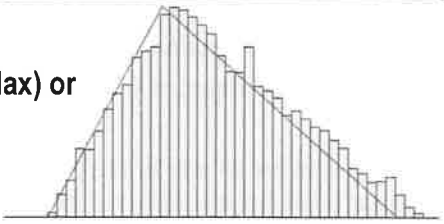


Typical uses are process times or move times. But be careful because it does have infinite tails. It probably should not be used unless the raw data indicates that the normal is the best fit. The normal distribution is not as common as it is generally believed to be.

7

Common Statistical Distributions

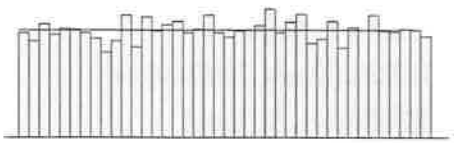
Triangular Distribution
TRIANGULAR(Min, Mode, Max) or
TRIA(Min, Mode, Max)



The triangular approximates many processes and may take on shape of many distributions. This distribution is used frequently because many times data is not available. However, there may be a best guess as to the most frequent value (mode), the smallest value (minimum) and largest value (maximum) for a given process. And of course it should be used if the raw data fits to this distribution.

Common Statistical Distributions

Uniform Distribution
UNIFORM(Min, Max) or
UNIF(Min, Max)



All permissible values are equally likely. This distribution would be used when there is very little data available or the raw data generated a uniform distribution. This distribution is sometimes used to assign discrete values to variables when each value is equally likely. This is executed by sampling between the upper and lower bounds of the discrete values and then rounding to the nearest integer.

Rockwell Automation

User Defined Distributions

CONTINUOUS(P1,V1,P2,V2,...)

– Pk = cumulative probabilities

– Vk = values of dependent random variable

Value returned is computed using linear interpolation within the range of two successive cumulative probability values. If a theoretical distribution could not be found that fits the input data, then one may use the empirical distribution of the data. This would be especially true when the process is known to take on a finite number of values, which would lead to the use of the discrete distribution below.

However, if the data is known or believed to be from a continuous valued input process, then it makes sense to interpolate between the observed data points to fill in the gaps. This distribution allows for that situation.

User Defined Distributions

DISCRETE(P1,V1,P2,V2,...)

– Pk = cumulative probabilities

– Vk = values of dependent random variable

Value returned based on cumulative probabilities specified

This distribution is sometimes used to assign discrete values to attributes or variables.

Fitting Distributions

Often, raw data will be available, but how can you use that data for your simulation study?

You can use distribution-fitting software (such as Arena's Input Analyzer) which will use the data and return the distribution that best fits that data set. Be careful, however, because most distribution-fitting programs will fit a distribution no matter how many data points you have. And they will fit a distribution even if the data truly does not represent any of the available distributions.

Now let us consider some words of caution when we start data collection and the distribution-fitting journey.

12

Number of Observations

First, you should have a sufficient number of observations to allow for a good fit. So what is a sufficient number? The number 30 is usually given as a good starting point. Also, the number of observations may depend upon what the modeler is able to collect. You should look at how the distribution corresponds to the histogram of the collected data.

13

Rockwell Automation

Dealing with Outliers

You should look at the histogram of the data during data collection for at least two reasons. If the distribution has outliers or if there are at least two peaks, then a single distribution is probably not representative of the data set.

Consider removing outliers and dealing with them in some other manner.

14

Does the distribution fit the process?

Also consider more than one distribution for this data.

When it comes to fitting data to a distribution, consider the origin of the data and determine whether the given distribution passes the "smell test" as well as the quantitative tests.

Only you or your system experts can determine if the distribution really represents what is going on in the system. You can also generate several variants from the distribution and plot another histogram to see if it is giving you what you desire.

15

Emulating versus simulating the system

Raw data can also be used to drive the model, which is sometimes done for model validation purposes. But for analysis, it is best to use the corresponding distribution or you will not get the randomness that is desired.

16

Arena Input Analyzer

Arena's Input Analyzer allows you to fit an Arena-supported distribution to a given set of data. To start the Input Analyzer, go to the Tools menu item and select the Input Analyzer feature.

Next, select the File > New option. Then select File > Data File > Use Existing.

You will then be asked to browse for the file containing the raw data observations. When browsing, the default file suffix that is "dst." Be aware that the raw data file may be a file with either a "txt" suffix or "dst" suffix. This file would have been created by saving the data into Notepad. The data may be exported from another source such as Microsoft Excel.

17

Arena Input Analyzer

Next, you select the data set. In this case, the data set is Input Analyzer Data Set and the histogram is created automatically.

Note the following data summary:

Data Summary

Number of Data Points = 36

Min Data Value = 1

Max Data Value = 30

Sample Mean = 13.3

Sample Std Dev = 9.55

Histogram Summary

Histogram Range = 0.999 to 30

Number of Intervals = 6

18

Arena Input Analyzer

You may then select the Fit option and select either a specific distribution or the Fit All that will give the best fit of the given data set.

By selecting Fit All, you will see the best fit of the given data.

You will also get a distribution summary as follows:

Distribution Summary

Distribution: Beta

Expression: 0.999 + 29 * BETA(0.529, 0.723)

19

Measures of Quality of Fit

- Besides "eyeballing" the fitted densities on top of the histograms, the Input Analyzer provides three numerical measures of the quality of fit of a distribution to the data.
  - Mean Square Error
  - Chi Square Test
  - Kolmogorov-Smirnov Test

- The first is the mean square error. The larger the square error value, the further away the fitted distribution is from the actual data and thus the poorer the fit. The Fit All Summary table orders the distributions from smallest to largest square error.

20

Measures of Quality of Fit

The Chi-square and Kolmogorov-Smirnov goodness-of-fit hypothesis tests are standard statistical hypothesis tests that can be used to assess whether a fitted theoretical distribution is a good fit to the data. Of particular interest is the Corresponding p-Value. To interpret this, larger p-values indicate better fits. Corresponding p-values less than about 0.05 indicate that the distribution is not a very good fit. Of course, as with any statistical hypothesis test, a high p-value doesn't constitute "proof" of a good fit, just a lack of evidence against the fit.

For more discussion of these issues, see Chapter 4 in the Simulation with Arena textbook.

21

Measures of Quality of Fit

Chi Square Test

Number of Intervals = 4

Degrees of Freedom = 1

Test Statistic = 1.17

Corresponding p-Value = 0.293

Kolmogorov-Smirnov Test

Test Statistic = 0.134

Corresponding p-Value > 0.15

• At this point, you can select Window > Fit All Summary, which will allow you to see the order in which the distributions are ranked according to the square error measure.

22

Fit All Summary

Function	Sq Error
Beta	0.0121
Uniform	0.0154
Exponential	0.0206
Erlang	0.0206
Triangular	0.0321
Normal	0.0368
Gamma	0.0457
Lognormal	0.0719
Weibull	0.0808

23

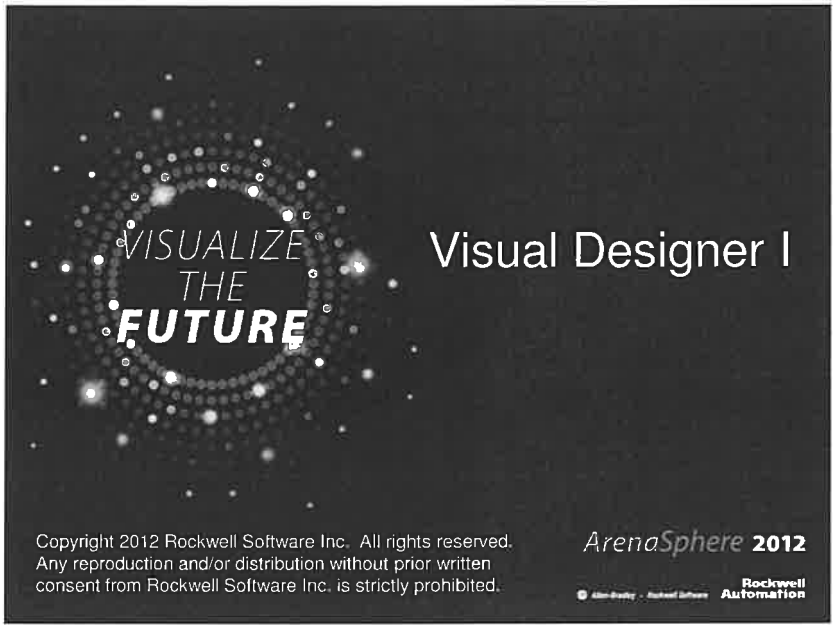
Rockwell Automation

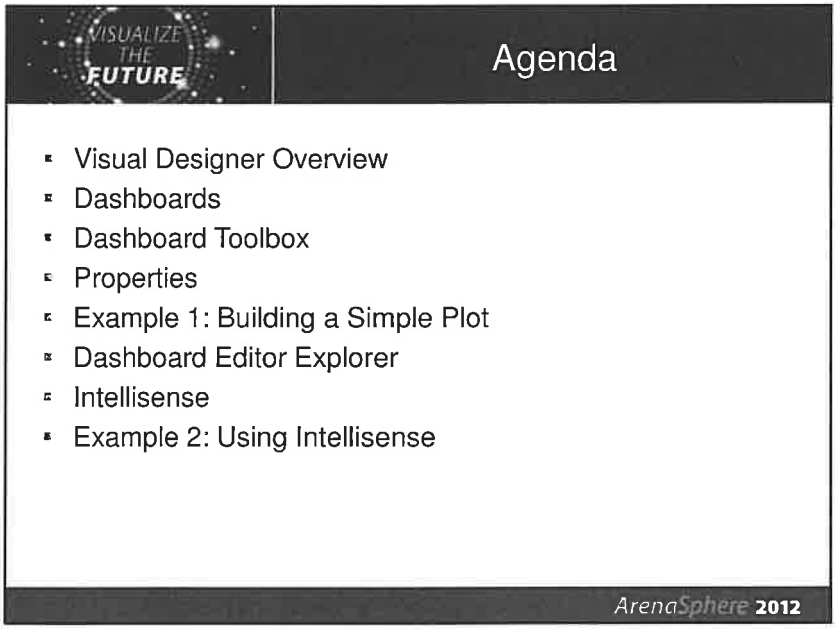
Now you have a distribution!

Once you have determined the best fit, you can select *Edit > Copy Expression which will copy the distribution expression into the clipboard* for pasting directly into the Arena model.

24

J: Visual Designer 1 - Dashboard


VISUALIZE
THE
FUTURE
Visual Designer I
Copyright 2012 Rockwell Software Inc. All rights reserved.
Any reproduction and/or distribution without prior written
consent from Rockwell Software Inc. is strictly prohibited.
ArenaSphere 2012
Rockwell
Automation

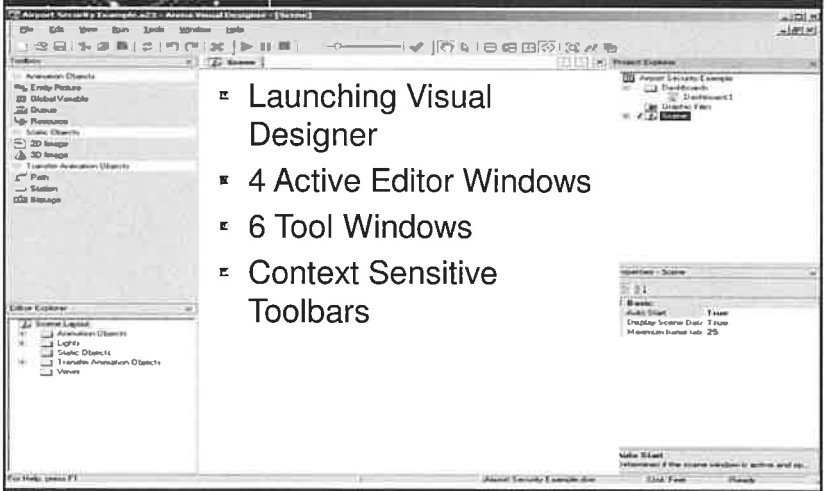

VISUALIZE
THE
FUTURE
Agenda
Visual Designer Overview
Dashboards
Dashboard Toolbox
Properties
Example 1: Building a Simple Plot
Dashboard Editor Explorer
Intellisense
Example 2: Using Intellisense
ArenaSphere 2012

Arena Visual Designer

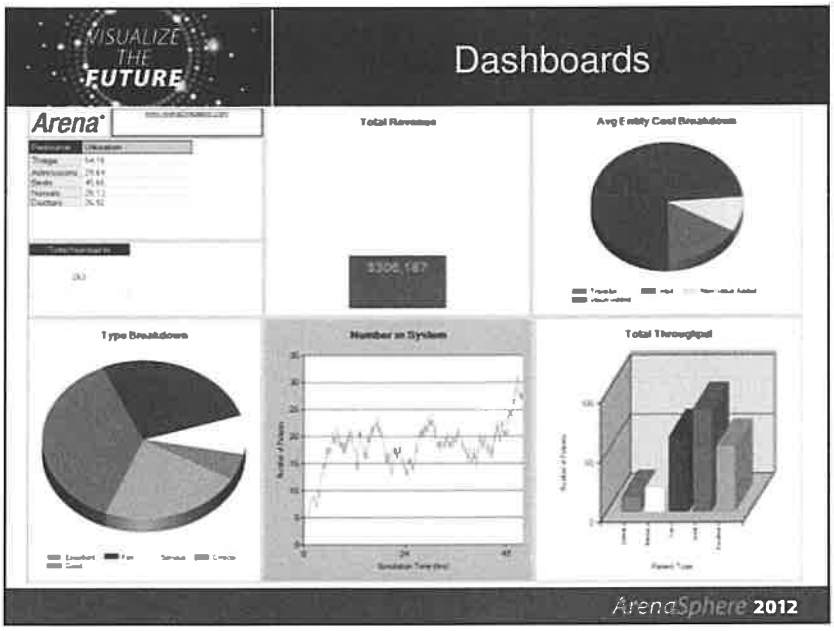


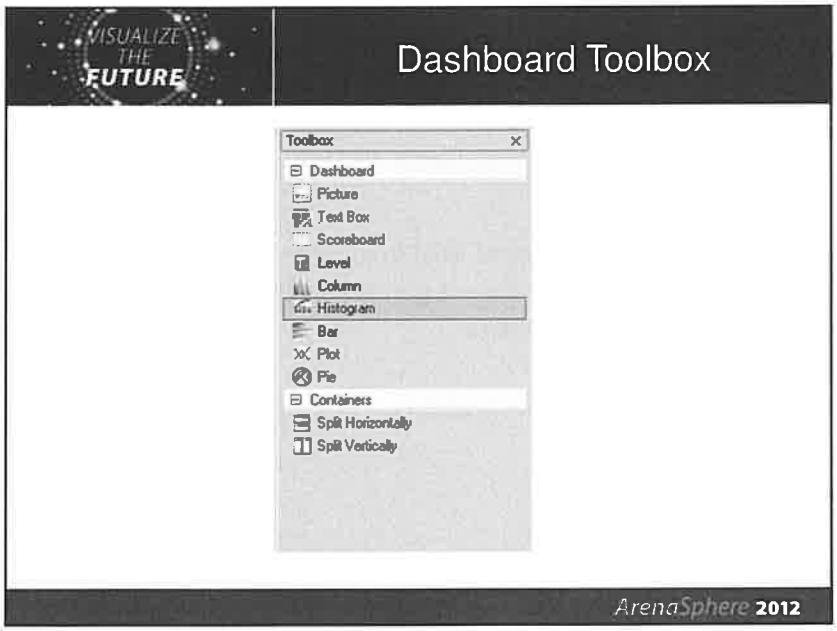
- New visualization environment within Arena
  - Dashboards
  - 3D Animation
- In process with Arena
- Beginning migration of user interface
  - NO changes to trusted SIMAN engine
  - NO changes to flow chart modeling paradigm

Visual Designer Environment



- Launching Visual Designer
- 4 Active Editor Windows
- 6 Tool Windows
- Context Sensitive Toolbars


VISUALIZE THE FUTURE
Dashboards
Arena
Total Revenue
Avg Entity Cost Breakdown
Type Breakdown
Number in System
Total Throughput
ArenaSphere 2012


VISUALIZE THE FUTURE
Dashboard Toolbox
Toolbox
Dashboard
Picture
Text Box
Scoreboard
Level
Column
Histogram
Bar
Plot
Pie
Containers
Split Horizontally
Split Vertically
ArenaSphere 2012

Properties



ArenaSphere 2012

Example 1: Building a Simple Plot

- Start with Airport Security Example.doe
- Open the Visual Designer application and create a new Dashboard
- Create a dashboard that contains
  - A Plot of the number in the "Check for Proper Identification" Queue
  - The Arena Logo
  - A Text Box with a description of your dashboard

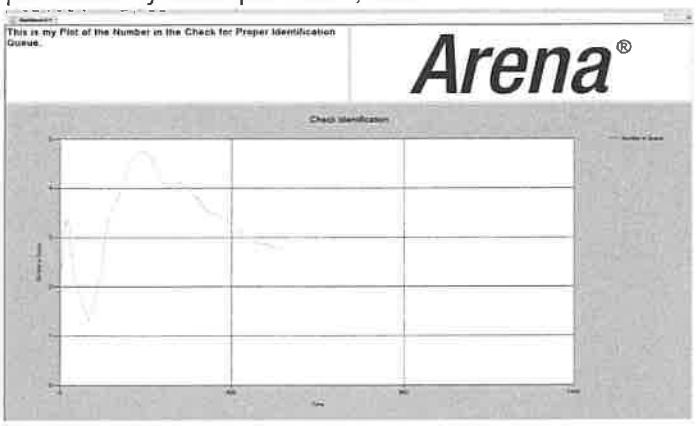
ArenaSphere 2012

Approach

- Use the Split Horizontally and Vertically to partition your containers
- Drag and drop a Plot Control into an empty container. Use the Properties and Data field to define the Check for Proper Identification.Queue
- Drag and drop a Picture control into an empty container Double-click to assign the logo image
  - The Arena Logo has been provided along with the training material
- Drag and drop a Text Box into an empty container and enter text to describe the project
- Explore the Properties to further enhance your controls

Solution

- Airport Security Example 1.doe, .a23



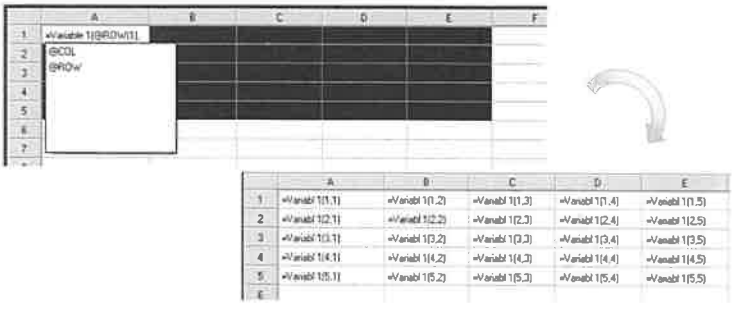
Dashboard Editor Explorer



ArenaSphere 2012

Intellisense

- Intellisense – a common tool used in programming to auto-complete expressions or lines of code quickly and efficiently
- Use with Dashboard Scoreboard control



ArenaSphere 2012

Example 2: Using Intellisense

- Add a new Dashboard and Scoreboard control
- Use Intellisense to animate the Scoreboard for every position of Variable 1
- Solution
  - Airport Security Example 2.doe, .a23

K: Supplemental Materials

Tips for Successful Practice of Simulation

by Deborah A. Sadowski & Mark R. Grabau

TIPS FOR SUCCESSFUL PRACTICE OF SIMULATION

Deborah A. Sadowski

Rockwell Software
333 Technology Drive, Suite 102
Canonsburg, PA 15317
dasadowski@software.rockwell.com

Mark R. Grabau

Accenture
11951 Freedom Drive
Reston, VA 20190
mark.r.grabau@accenture.com

ABSTRACT

Succeeding with a technology as powerful as simulation involves much more than the technical aspects you may have been trained in. The parts of a simulation study that are *outside* the realm of modeling and analysis can make or break the project. We explore the most common pitfalls in performing simulation studies and identify approaches for avoiding these problems.

1 INTRODUCTION

If you've been charged with performing a simulation study, you may not know it, but you have quite a challenging job on your hands. (Be sure to share this with your boss for a bit of professional sympathy or, if you're lucky, a well-deserved raise.) To succeed with the task you've been assigned, you'll need an incredible combination of both quantitative and qualitative skills, a lot of support from many areas of your organization, and a suite of tools and techniques to enable your work.

Much has been written and researched in the last area, resulting in a mind-boggling array of modeling/analysis techniques and software for performing simulation studies. Your selection of the methodology you'll use and the supporting tools to do the work will contribute to the success of your project.

Regardless of the approach you're taking and the product you're using, though, succeeding with simulation requires much more. As it turns out, many of these supplementary aspects can influence your likelihood of doing a good job more greatly than the traditionally discussed methods and tools available to you.

2 WHAT DOES "SUCCESS" MEAN?

First, it's important to identify how success will be defined. In the best scenarios, a successful simulation project is one that delivers useful *information* at the appropriate *time* to support a meaningful *decision* (see Figure 1).

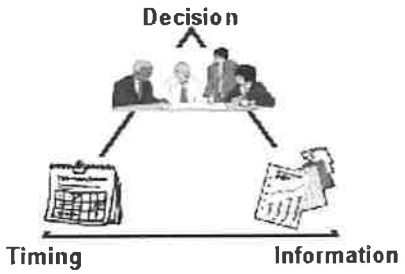


Figure 1: Elements of Success

2.1 The Right Information

The most important aspect of presenting the right information is to look at it from the perspective of the audience for which it's intended. Think about *what* they need to know and *why* they need to know it—in the context of what they're going to do with this information to deliver value to your business. Try to anticipate the questions they might ask. And remember, their view of the system you're studying is bound to be different from yours, so take the time to introduce them to how you look at things, and explore how this differs from their perspective. You may need to create more elegant animations or design custom reports, for example, to communicate what you're doing and what you've learned easily and effectively.

The nature of the information that's needed may vary, even at different times within a project. When you embark on your simulation journey, one of your first tasks should be to define this as explicitly as possible, keeping in mind that there are many constituents who may have an interest in your work.

Those who make and influence decisions will certainly be interested in the data that's typically associated with a simulation study (cycle times, costs, resource utilizations, etc.). Beyond this obvious information, though, may lie other types that are equally important. Animation of the simulated process might be key to the success of the study, by supporting model validation and influencing decision-

makers. Options that were unexplored or quickly discarded should be documented so that others know both what you did (or didn't do) and why you made those decisions.

You also should trust that instincts you've developed about the system might be as or more valuable than those of the "experts" with whom you've consulted. One of the great benefits of assigning someone to perform a simulation study is the perspective they have on the system. While the people who are performing or implementing a process/system have a mastery of great detail concerning their area, the simulation analyst gains a higher-level overview—while still understanding the process at a fair level of detail.

2.2 The Right Timing

The timing of when you are able to deliver meaningful information also is critical to a project's success. A high-fidelity answer that's too late to influence a decision isn't nearly as good as a rough-cut estimate that's in time to help.

Note, too, that this applies throughout a study, not just at its completion. If you can provide preliminary insights into a system's behavior early in a project, the owners of the design might change the options they consider or adjust the focus of the simulation efforts.

2.3 The Right Decision

The third aspect of succeeding may be out of your control, but is important for you to understand. Namely, for the project to succeed from your efforts, you need to influence an important decision. Wonderful simulation work, advanced analysis, and eye-grabbing animation, all completed on-time still are of no value if they aren't delivered to the *right person* in the *right context*.

If you're adept at corporate politics, you'll probably find it easy to figure out who the right people are and to what extent they overlap with those who've been identified as important decision-makers. (Surprisingly, sometimes these lists differ.) If not, then as your project moves forward, you should work to identify where the power is and tailor your communications to put the right information in front of these individuals.

When you present any information from your study, you also must consider the decision environment: where are the sensitive areas, what preconceptions exist, and what's really important. Many simulation studies are initiated to "prove" that a planned course of action is right. If your analysis concludes to the contrary, then the environment you'll be walking into will be significantly more challenging than cases where there are open minds. In such a situation, which you should try to discover early in the project, you should allocate extra time for additional analysis after your project is "complete," just in case you find out that the existing wisdom was misguided and need extra ammunition to make your case.

3 WHERE COULD YOU GO WRONG?

To reach this goal of succeeding with simulation, two questions seem most relevant, namely how can things go wrong and what should you do to make them go right? As often happens, while you may not have control over all of the relevant aspects of your work, it can be invaluable to understand the importance of them as you plan your efforts and design your reports and presentations.

3.1 Tackling the Wrong Problem

Sometimes, the biggest mistake is made at the outset of a simulation study. If your organization or client has picked the wrong problem to explore with simulation, you might be put at a high risk of failure before you've made your first mouse click.

One of the interesting places where this occurs is when simulation is used to prove what's already "known" or decided. In some firms, certain thresholds of capital acquisition require a simulation study, though it might be initiated well after firm commitments to a particular course of action have been made. If you're charged with one of these types of projects, you're either really lucky (i.e., the plan that's already in place is right) ... or you're in a very dangerous situation. Unfortunately, you won't know which is the case until you complete your work.

Particularly because of the animation that accompanies most simulation studies, another danger presents itself when identifying candidate projects. Once an analyst becomes adept at performing simulation studies, he/she can fall into the trap that "when you have a hammer (simulation), everything looks like a nail (career- or business-enhancing opportunity)." Certainly, many problems require simulation; the needed decisions can be effectively made only by looking at them through the perspective of a simulation analysis. However, other problems can be readily solved using other tools, such as queuing analysis, optimization, or simple spreadsheet calculations. When you're about to embark on a simulation project, step back and double-check that simulation's the best tool for the job. If something simpler can provide the same quality of results, then avoid the cost of the simulation study and use the appropriate tool.

The most common types of misguided simulation studies, though, are those where the scope is too ambitious or ill-defined. It's difficult to figure out where the boundaries should be in a complex system, since often it seems that everything could be an important factor on performance. You must work hard early in a project to discover what to exclude from the study; while it's hard to say "no," it can be critically important to be willing to do so.

3.2 Working on the Right Problem ... at the Wrong Time

To increase the chance of providing a good answer at the right time, you may need to think carefully about when to start a simulation project and whether to put the brakes on, even if you've established momentum. If the designers of the system/process are still considering widely differing ideas, or are brainstorming for how to solve some of the fundamental problems in the system, then it may be premature to perform more than a rudimentary analysis.

It's more difficult to identify timing problems once a project is under way. If there are regular and significant changes to the nature of the project, you'll feel the effects since you'll have to rework your model. It's hard, though, to know whether or when you should pause the simulation work to let the project team do some more preliminary design. Or, it may be that your best value is to use simulation for very rough-cut analysis, putting on hold the more detailed study that initially was chartered.

There's also danger on the other side of the timing spectrum, where a simulation study is started too late to be successful. This often begins with a panicked call from a project manager, who says that he/she "absolutely must have a simulation done starting now!" Of course, the natural response is to bring simulation to the rescue! Unfortunately, the cavalry riding over the hill are seldom carrying laptops with completed, validated, and verified simulation models.

If you're presented with a request like this, you should carefully lead the project manager through what is feasible to be of value. Often, this will involve intense negotiating, where the project manager wants a detailed, thorough analysis done quickly and the simulation analyst must focus on what's can reasonably be done so that the project starts out with attainable goals.

3.3 Missing the Warning Signs of the "Data Woes"

Ask any experienced simulation analyst what the most aggravating, challenging, dangerous aspect of a project is, and you're likely to hear "data" in reply. According to Ricki Ingalls, manufacturing strategy manager for Compaq Computer, "It's the data management that takes up most of the time ... getting the data, running it, then analyzing it" (Andel 1999).

The data woes are somewhat analogous to the story of Goldilocks and the Three Bears: you can have too little, too much, or just the right amount ... and still find yourself in trouble.

3.3.1 Too little data

Most often, if there are problems with data, it's a lack of information. Service times, yield probabilities, defect rates, rework percentages, and many other important aspects of a system's dynamics may not be collected for other business purposes. Because getting this data can be very time-consuming, it's critical to establish your data needs as early in a study as possible and to assess whether the data exists immediately. Your estimated project duration might vary by 100% or more, based on what you find when you go looking for numbers.

3.3.2 Too much data

In your search for data, you may find the opposite problem, that there's far too much of it. While the particular information you need may exist, even perhaps electronically in a database or spreadsheet, you may spend days trying to locate it amidst all the other data that's with it.

In this circumstance, it's imperative to find help from someone who is knowledgeable about the data and whom you can educate about your exact needs. Depending on how the data will be used, you may also need some help from IS in extracting the data. For example, if you'll be using a tool that fits distributions to data for use with simulation, then you'll probably need to transfer the data to an intermediate form. If so, ask for these resources early so that they don't become an impediment to keeping your project moving forward on-schedule.

3.3.3 Just the right amount, but what does it mean?

Sometimes, you may actually run across a case where the amount of data closely matches your needs. When this happens, it'll look like smooth sailing.

However, when you look at the data, be sure to understand what it *really* means. What you think of when you say cycle times, for instance, may be very different from the data that's stored in a cycle-time table, which might include waiting time, breakdown times, and other properties that are modeled separately in simulation.

3.4 Letting the Window of Opportunity Close

The greatest and most widely discussed risk of failure with simulation is that you won't finish the job on time. Creating a valid, useful simulation model is essentially a software development project, with similar risks and challenges. Returning to our definition of success, the timing of the information can be as important as the quality of the information—if the decision's made before you start your analysis, you might as well archive your files and move on to something else.

There are myriad reasons why simulation projects are late in delivering results. Four particular pitfalls seem worth special consideration.

3.4.1 Getting lost in detail

One of the easiest traps to fall into is getting hooked on modeling. The art of simulation involves assessing what level of detail is required to support the project's goals. It's tough to do this right, though, because often you can't tell whether the detail is needed until you've developed it; and once the work is done, it's hard to justify removing it if it's unimportant.

Whenever possible, err on the side of keeping the model simple, unless you have the luxury of significant slack in your schedule. It's usually more important that you are able to perform some level of analysis of a system in a timely fashion than to run the risk of having no results to deliver when they're needed.

To avoid this trap, find a colleague who can frequently listen to you review your work. For most projects, it's worthwhile to sit down with this person at least two or three times a week, usually for only 15 minutes if you organize the discussion well. Sometimes, the time will be worthwhile for the insights *you* find by having to explain what you're doing to someone else, even if he/she doesn't directly question or contribute anything.

If the detail is being driven by those external to the project (e.g., decision-makers, clients), then carefully prepare an explanation of the risk to the project schedule associated with the added detail. If possible, perform a sensitivity analysis on the area in question; if significant modifications (e.g., 15-30%) to the process don't significantly change your decision variables, then it's unlikely that capturing the fine nuances of the system will be meaningful.

Sometimes, the need for additional model fidelity is driven by a desire for more realistic animations. While many simulation analysts view this as of minor value for meeting their goals, the quality of the animation can be very important in effectively communicating project recommendations. Once again, early planning and communications can be critical, so that you can build a reasonable project schedule that incorporates sufficient time for all aspects of the effort.

3.4.2 Leaving analysis for the end

There's a common misconception that performing a simulation study involves a sequence of steps (e.g., project definition, model formulation, verification, validation, analysis). To the contrary, all elements of a simulation project should be performed repeatedly throughout the effort, growing in scope as the model progresses.

In the traditional view, projects suffer from too strong a focus on the model (and perhaps the animation), so that after the inevitable delays and problems, there's no time left to run scenarios. Instead, the analyst is faced with a presentation deadline that's firm and little time to experiment, analyze, or think.

Instead, you should schedule the project in complete phases. Intermediate milestones, spaced no more than about two weeks apart in a medium to large project, should include specific goals for the model, animation, data, and analysis. By the time you reach the last 25% of your time on the project, you should have addressed the basic analysis issues of run length, warm-up time, etc. and should already have performed preliminary analysis on the model for a number of different scenarios.

3.4.3 Having too much fun with animation

If you've ever used PowerPoint, you'll understand this concept. At the point where you've drafted all of your content (i.e., the important stuff), you probably feel like you're almost done. Then it's time to adjust the fonts and slide transitions, to add figures and clip-art, and to twiddle with the custom animations. Surprisingly, you might find that the "prettying up" of your presentation takes more time than drafting the materials in the first place!

Animation holds a similar attraction in simulation studies. With the mouse in-hand, you can easily fall into the trap of adjusting, tweaking, and enhancing the animation far beyond what's needed, just because of its entrancing nature. (Some readers may not identify with this; if so, consider yourself fortunate!) As with many other addictive behaviors, the only treatment for endless animation is a recognition of the risks and discipline in your work.

3.4.4 Testing at the end of the project

As with analysis, verification of the model must be performed throughout a project. Because it's even more tedious and uninteresting than analyzing the simulation, it's easy to leave testing for late in the project.

Don't do it! Simulation can be a powerful influence on decision-makers. If you reach conclusions based on a faulty model, you've done a greater disservice to your organization than if the simulation had not been performed at all.

Think about a test plan early in the project and revise it as you progress. Delivering a "state of the model" assessment, which grades the various segments of the model regarding quality and completeness, should be a standard part of your simulation project at each milestone.

4 HOW CAN YOU SUCCEED?

With all of these challenges, it's a wonder that anyone could possibly do all this! But there are a few simple habits you can develop that can boost your likelihood of success substantially.

4.1 Establish a Clear Focus

The success or failure of a simulation study begins with establishing a reasonable scope and the subsequent planning of the project. The study's specification must be formalized by obtaining sign-offs on focused objectives. Failure to obtain these commitments, or establishing objectives that are too vague, can set the project up for failure before it begins.

In many cases, the decision-maker or project initiator—we'll refer to him/her/them as "the client," though they're often internal to your organization—doesn't know what to expect from the simulation team. Likewise, often the simulation team doesn't know what is expected of them. It's extremely important that the client and the simulation team agree and adhere to the scope of the project. If the scope changes, these modifications must be agreed upon and their impacts acknowledged by the client and simulation team. The simulation project manager must obtain schedule and cost relief if additional resources, time, or personnel are required to complete the simulation on-time and according to the new specifications.

The project should also encompass animation at the appropriate level of detail. The animation demonstrates an understanding of the system being studied and provides credibility with senior management; therefore, it should always be a consideration while planning the scope and schedule for the project. Many times, the client may even be less interested in the actual results; but if it the animation looks impressive, at a minimum, the simulation team gains instant credibility with any output results that follow.

4.2 Plan Carefully and Thoroughly

Once the project scope has been established, as mentioned earlier, you should identify data requirements. The earlier that you get a handle on what data is needed, the sooner data gathering may begin. As any simulation professional knows, data collection and analyses often take longer than building the model itself.

Given the objectives of the project, prototypes of the output reports may also be designed and agreed upon so that the modeler is sure to calculate the appropriate statistics during the simulation. Unnecessary amounts of rework can be avoided if the modeler knows ahead of time what statistics need to be collected. The difference can even be as simple as only having to use a different modeling construct; e.g. a Batch versus a Match module in Arena®.

It's also imperative that you plan the project at an appropriate level of detail. The project manager should develop a project plan using a work breakdown structure. This approach is characterized by breaking major tasks into smaller ones with task times and the resources required to complete each task.

Examples of major tasks that should be accounted for in the plan are:

- gathering and analyzing data,
- building the model,
- animating the model, and
- analyzing the output.

An example of a breakdown for gathering and analyzing data would be:

a) gather arrival data,
b) gather process time data,
c) analyze arrival data,
d) analyze process time data.

The manager should also schedule milestones for each of these major tasks, in addition to milestones for subtasks. The schedule must be continuously reviewed by the project manager and should also be reviewed by project members. The client must be given frequent updates concerning the schedule: how the project is proceeding and whether or not milestones are being met.

4.3 Build a Realistic Timeline

Typical downfalls to avoid in managing simulation projects include underestimating time for three main tasks:

- data collection,
- verification, and
- performing analysis runs.

As mentioned earlier, data collection inevitably takes longer than anticipated. If the schedule is getting tight, another pitfall is to do less verification or to only do a top-level verification. Particularly in cases where the model includes complex logic, or where the model scope or system design changed significantly during the project, the process of verifying that the model actually does what you think it's supposed to do (a/k/a debugging) can be tedious and very time-consuming. When you create the project schedule, you should be sure to build verification time in at regular intervals; for a large project, as much as 20% to 30% of the time set aside for modeling should specifically be assigned as testing and debugging time. If appropriate time is not allotted for verification, the time may "bleed" into time that was allocated for analysis runs.

Until the first true analysis run is performed, the computing time required for a run is unknown, which still does not account for the number of replications, a warm-up period for a non-terminating system, and multiple design points in an experiment, let alone the appropriate analyses to determine all of the above. In an effort to finish the project on schedule, the impulse reaction is to do fewer or shorter runs, which may produce invalid results. In addition, often not enough time is allowed for an appropriate sensitivity analysis of the system in order to supply the client with the value the simulation project intended to provide. The irony is that the results of the simulation are the reason the project was undertaken in the first place!

To avoid this pitfall, design the details of your analysis at the first opportunity, no later than when the model is about 75% complete. Then reassess the time you've set aside for final analysis, based on how long your runs will take, as well as the new perspectives you have on how many scenarios you'll be evaluating. In projects where you'll be performing optimization analysis, this would also be a good time to begin refining the number and search ranges for your controls (input variables) and to perform some initial optimizations to predict how long they'll take when the model is completed.

4.4 Constantly Review and Reassess

The model and other deliverables should be reviewed early and often, with more intensity as the project nears completion. Structured walkthroughs with colleagues and clients are ideal for discovering problems with logic or errors in the model. In a structured walkthrough, the modeler steps through the modeling constructs and explains model logic, how areas were abstracted, and what assumptions were made. Colleagues may point out easier ways to accomplish portions of the model or even point out an incorrect interpretation of the system under study. In addition, the simulation team should review the model specifications, data analyses, animation, output reports, and client presentations.

Throughout the duration of the project, flexibility is important. As situations arise such as scope changes, problems with the data collection, or lack of subject matter expert availability, the simulation team must look for new ways to solve problems or work around them. More importantly, as these situations arise they should also be questioned regarding whether or not they are consistent with the true motivation for the project.

5 WHAT CAN YOU DO NOW?

While all of this may sound good "for my next project," there are many tasks that may be accomplished immediately. If your simulation methodology doesn't already have one, draft an outline for a typical simulation project and use this as a template for the work breakdown structure of current and future projects (see Kelton, Sadowski, and Sadowski 1998). Condense current and future project goals to reflect easy-to-remember slogans such as, "It's the cycle time stupid" in order for the entire team to remain focused on the goals. Conduct a peer review, or structured walkthrough, as soon as an opportunity presents itself. Finally, consult some of the references listed in this paper to gain additional insights into how to succeed as a simulation analyst.

REFERENCES AND ADDITIONAL READING

Andel, T., 1999. Get it right before it's real. *Material Handling Engineering*. Penton Media, Inc. Cleveland, OH, USA.

Banks, J., J. Carson, B. Nelson, 1996. *Discrete-Event System Simulation.* Prentice-Hall, Upper Saddle River, NJ, USA.

Banks, J., 1998. Plan for success. *IIE Solutions*. Institute of Industrial Engineers. Norcross, GA, USA.

Brunner, D., et al., 1998. Toward increased use of simulation in transportation. In *Proceedings of the 1998 Winter Simulation Conference*, ed., D.J. Medeiros, E. Watson, M. Manivannan, and J. Carson, 1169-1175. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Ferrin, D. and R. LaVecchia, 1998. Customer interfacing: lessons learned. In *Proceedings of the 1998 Winter Simulation Conference*, ed., D.J. Medeiros, E. Watson, M. Manivannan, and J. Carson, 1347-1350. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Kelton, W. D., R. Sadowski, and D. Sadowski, 1998. *Simulation with Arena*. McGraw-Hill, New York, NY, USA.

Law, A. and W. D. Kelton, 1991. *Simulation Modeling and Analysis*. McGraw-Hill, New York, NY, USA.

Profozich, D., 1998. *Managing Change with Business Process Simulation*. Prentice-Hall, Upper Saddle River, NJ, USA.

Rohrer, M. and J. Banks, 1998. Required skills of a simulation analyst. *IIE Solutions*. Institute of Industrial Engineers. Norcross, GA, USA.

AUTHOR BIOGRAPHIES

DEBORAH A. SADOWSKI is a consultant with Rockwell Software (formerly Systems Modeling). Deb has held many roles at Systems Modeling, including Vice President of Development during the creation of Arena. She is co-author with W. David Kelton and R. P. Sadowski of the textbook, *Simulation with Arena*. Deb received her B.S. and M.S. degrees in Industrial Engineering and Operations Research from The Pennsylvania State University. She presently represents the IEEE Computer Society on the WSC Board of Directors.

MARK R. GRABAU is an executive in Accenture's Decision Integration practice. He has over 9 years of experience applying simulation modeling on consulting interventions in the transportation, pharmaceutical, telecommunications, manufacturing, and government industries. He earned his B.S. in Operations Research from the United States Air Force Academy in 1992. He earned his M.S. in Operations Research and his M.S. in Statistics from the Colorado School of Mines in 1997.

L: New Features in Arena 15 & 15.1

What's New?

The following are included in Arena 15.0:

Arena 64 Bit

Arena 64 bit takes full advantage of the memory available with 64 bit processing. 32 bit applications are limited to 2 GB of memory—regardless of the amount of memory installed on the computer. 64 bit applications can utilize up to 8 Tb of memory. This means that significantly larger models can be built in Arena and not require page swaps.

Arena 32 and 64 Bit

The side by side install of Arena 32 and 64 bit on the same computer is not supported.

Arena 64 Bit Activation

Arena 64 bit currently requires the use of a WIBU dongle for activation. You will need to obtain a WIBU dongle in order to run Arena 64 bit with an Arena license. Contact Arena tech support to find out how to obtain a WIBU dongle. A future release of FactoryTalk Activation Manager will allow activation of Arena 64 bit without a dongle.

Animated Network Links

Network links have been enhanced in the transfer animation bar. Network links can now display the current state of intersections and zones in the animation. Available states are unoccupied, occupied and captured. This new feature makes it easier to evaluate and analyze network simulations.

SMARTS Update

Links to the SMARTS files are now available in the online help. SMARTS files provide examples of key Arena capabilities. They are a great tool for learning how to use Arena.

Below are four new smarts illustrating the use of Arena for Agent Based Modeling. To access these files in Arena, use the File menu select "Browse SMARTS...." and select one of the following models located in the Agent Based folder:

- Agent Based Basic Movement.doe: Illustrates how to model movement in a coordinate space.
- Agent Based Bug Spray.doe: Illustrates how to model bugs in a coordinate space and the effect of bug spray in the system.
- Agent Based Habitat Utilization.doe: A model of bugs wandering through a coordinate space (habitat) eating food and moving around to eat more.
- Agent Based Reproduction.doe: A model of the reproduction of bugs in a coordinate space.

New Functionality

- **Entity Attributes**

  Entity Attributes will display in alphabetical order.

- **User-defined attributes**

  User-defined attributes will display in alphabetical order at runtime at the same level in the attribute tree as the Attributes.

- **Changing module connector colors**

'Set Colour' was added to the right-click menu to change the colour of Module connectors.

- **Links to SMARTS**

  A link will be provided to the SMARTS files from the Help Menu.

- **Dynamic Batch Mode**

  A 'Batch mode toggle' toolbar button will be added to the Run Interaction toolbar to allow the batch mode feature to be turned on and off during a simulation run.

- **Highlight Active Module in Batch Mode**

  Highlight active module will be enabled when running in batch mode.

- **Highlight Active Module Toolbar Button**

  A "Highlight Active Module" toolbar button will be added to the Run Interaction toolbar.

New Modules

- **Clone Module**

  This module will provide the ability to make multiple duplicates of an entity in one module.

  A new smart describing the use of the new Clone module has been added. To access this file from Arena, use the File menu select "Browse SMARTS...." And select the model located in the Basic Concept folder:

  - Basic Concept Making Copies of Entities.doe: Illustrates the use and differences of the Separate and Clone modules to make copies of an entity.

- **GoTo Label and Label**

  These modules are used for directing model flow to a specified label.

- **Picture Data Module**

  This module will be used to define one or more Entity Picture(s) in the system.

- **Adjustable Batch Module**

  This module is similar to the Basic Process Batch module, with addition of logic to allow the creation of partial batches based on time or based on a signal sent by user defined logic.

  Below are two new smarts illustrating the use of the new Adjustable Batch module. To access these files in Arena, use the File menu select "Browse SMARTS...." and select one of the following models located in the Batching folder:

  - Batching Using Adjustable Batch with Max Wait Time.doe: Illustrates the use of the Adjustable Batch module using the 'Max Wait Time' Partial Batch Method. A maximum time is specified to wait for the desired optimum batch size.
  - Batching Using Adjustable Batch with Signal.doe: Illustrates the use of the Adjustable Batch module using the 'Signal' Partial Batch Method. A signal value is specified that can be used to trigger a partial batch if the optimum batch size is not yet reached.

- **Assign Attribute Module**

This module will be used to assign a value to one or more attributes of either the current active entity or other entities in the system.

A new smart describing the use of the new Assign Attribute module has been added. To access this file from Arena, use the File menu select "Browse SMARTS...." And select the model located in the Advanced Concept folder:

- Advanced Concept Change Attribute with Assign Attribute.doe: Using the Assign Attribute module to change the value of an entity attribute.

- **Search and Remove Module**

The Search and Remove module searches a queue or a temporary group (batch) to find the entity or entities that satisfy the specified search condition, removes them from the temporary group or batch and then sends them to a designated module.

Below are several new smarts illustrating the use of the new Search and Remove module. To access these files in Arena, use the File menu select "Browse SMARTS...." and select one of the following models located in the Advanced Concept folder:

- Advanced Concept Search and Remove Entity Batch.doe: Using the Search and Remove module to search all members of an entity batch and remove the entitiy with the maximum search condition.
- Advanced Concept Search and Remove Q Maximum Condition.doe: Using the Search and Remove module to search all members of a queue and remove the entity with the maximum search condition value.
- Advanced Concept Search and Remove Q Sets.doe: Using the Search and Remove module to search all members of a queue set and remove entities that satisfy a search condition.

- **Station Data Module**

This module will be used to define one or more stations in the system. This module will facilitate the ability to define stations through the use of the spreadsheet data interface, instead of the Station logic modules dialog interface.

The station number property is exposed here to allow the users to reference a group of like stations without having to create a Station set.

Statistics Template Panel

A new template panel has been added to Arena which includes several new modules for collecting model statistics.

- **Timestamp module**

The Timestamp module assigns the current simulation time to the specified Attribute Name.

- **WIP module**

The WIP module increments or decrements the specified WIP variable by simulation time to the specified Attribute Name.

- **Tally Statistic Data Module**

The Tally Statistic Data module is used to specify parameters for tallies that are used to collect observational statistics. The values recorded for the Tally Statistic are specified in the Basic Process Panel's Record module.

- **Counter Statistic Data Module**

  The Counter Statistic Data module is used to specify parameters for counters that may be used to keep integer count statistics on events occurring in the model.

- **Time-Persistent Data Module**

  The Time-Persistent Statistic Data module is used to obtain time-persistent statistics in a discrete system. Statistics can be obtained for any Arena user variable, system variable, or expression containing user or system variables.

- **Output Statistic Data Module**

  The Output Statistic Data module defines Arena expressions whose values are to be reported in the Arena Summary Report and optionally recorded in output files at the end of each replication of a simulation.

- **Frequency Statistic Data Module**

  The Frequency Statistic Data module is used to record the time-persistent occurrence frequency of an Arena variable/expression or resource state.

Below are two new smarts illustrating the use of the new Statistics panel modules. To access these files in Arena, use the File menu select "Browse SMARTS...." and select one of the following models located in the Statistics folder:

- Statistics Timestamp and Record modules.doe: Illustrates the use of the Timestamp and Record modules to collect observational (time interval) statistics.
- Statistics WIP module.doe: Illustrates the use of the WIP module to track the number of entities in process for a given area of a model.
-

Module Updates

- **Record Module**

The Record module has been improved to enable the user to define multiple statistics with one module instance.

- **Attribute, Variable and Expressions Data Modules**

These modules have been improved to enable the user to include a comment to describe the use of each of these elements for model documentation purposes.

- **Match Module**

The Match module has been updated to include optional batching of the matched entities after the match is made. The batches of entities can be permanently or temporarily grouped.

A new smart describing the use of the updated Match module has been added. To access this file from Arena, use the File menu select "Browse SMARTS...." And select the model located in the Batching folder:

- Batching Using Match with Batch option.doe: Customer order and processing model utilizing the 'Batch Action after Matching' option.

- **Pickup Module**

The Pickup module now includes the ability to specify the queue in one of four ways: as a queue symbol name, as a specific set member, as an expression, or using an attribute symbol name.

- **Read/Write Module**

The ReadWrite module includes functionality to make using CSV (comma separated value) input and output data files easier to use.

Below are two new smarts illustrating the use of the updated Read/Write module. To access these files in Arena, use the File menu select "Browse SMARTS...." and select one of the following models located in the External Data Files folder:

- External Data Files Reading a CSV File.doe: Reading from a CSV (comma separated value) file. Note that there is an accompanying "simdat.csv" data file.
- External Data Files Writing CSV File with Header.doe: Writing to a CSV file and including a header row.

- **File Data Module**

The File module includes functionality to make using CSV (comma separated value) input and output data files easier to use.

- **Transporter Data Module**

The transporter data module now includes a new type option called forklift.

- **Elements Template Panel Modules**

The Elements template panel modules have been modified to include the spreadsheet view.

What's New?

The following are included in Arena 15.1:

Windows 10 Support

Arena will now officially support the Windows 10 operating system. Arena will no longer officially support the Windows Vista operating system.

New Scoreboard Animation

A new Scoreboard Status Display has been added to the "Animate" toolbar. This animation object control can be used to display model variables and expressions similar to a spreadsheet. Cell contents can display descriptive text or the current value of a model variable or expression.

A new smart illustrating the creation of a Socreboard animation has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the Animation folder: **Animation Arena Scoreboard.doe**

New Modules

- **Gather Module**

  The **GATHER** module is used when an entity requires one or more units of any one or more members of a resource set of interchangeable resources. Unlike the SEIZE module, GATHER allows an entity to seize units from multiple members of a specified resource set. SEIZE, when specifying to SELECT from a set, can only seize the specified units from one member of the specified set.

  A new smart illustrating the differences between the Gather module and the Seize module has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the Advanced Concept folder: **Advanced Concept GatherVSeize.doe**

- **Disperse Module**

  The **DISPERSE** module is used when an entity is ready to release some or all of the units of the resource set members that were seized using a GATHER module.

  A new smart describing the use of the new Gather and Disperse modules has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the Advanced Concept folder: **Advanced Concept GatherDisperse.doe**

Module Updates

- **Set Data Module**

  The Set module has been improved to enable the user to specify a data file and corresponding recordset to utilize direct read to define the set members property. In addition, the user can now define resource sets using other resource sets.

  A new smart illustrating the use of the new resource set of sets has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the Resources folder: **Resources SetHierarchy.doe**

  A new smart describing the use of the new direct read feature has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the External Data Files folder: **External Data Files Sets Direct Read from Excel.doe**

- **Advanced Set Data Module**

The Advanced Set data module has been improved to enable the user to specify a data file and corresponding recordset to utilize direct read to define the set members property.

- **Conveyor Data Module**

The Conveyor data module has been improved to enable the user to specify a data file and corresponding recordset to utilize direct read to specify the conveyors numeric properties.

A new smart illustrating the use of the direct read feature has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the Conveyors folder: **Conveyors Data DirectRead.doe from Excel**

- **Segment Data Module**

The Segment data module has been improved to enable the user to specify a data file and corresponding recordset to utilize direct read to specify a segments' repeatable *Next Stations* data.

A new smart describing the use of the direct read feature has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the Conveyors folder: **Conveyors Data DirectRead from Excel.doe**

- **Schedule Data Module**

The Schedule data module has been improved to enable the user to specify a data file and corresponding recordset to utilize direct read to specify a schedules' repeatable *Durations* data.

A new smart describing the use of the direct read feature has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the External Data Files folder: **External Data Files Direct Read from Excel for Schedule Data.doe**

New Functionality

The ReadWrite module has been modified to include two additional format options. Real decimal (%f) and real scientific notation (%e) have been added to the existing options of real(%g), integer (%i) and string (%s).

A new function named "ExcelDatetoBaseTime" function has been added. This function will convert the Excel date serial number to base time units in Arena. A new smart describing the use of the new function has been added. To access this file from Arena, use the File menu, select "Browse SMARTS...." And select the model located in the External Data Files folder: **External Data Files ExcelDateToBaseTime function.doe**

Two new resource functions named "LESR" and "LERR" have been added. The function LESR will return the last entity that seized or gathered units of a specified resource. The function LERR will return the last entity that released or dispersed units of a specified resource.

New Smarts

The following example models have been added to the Smarts file library:

-**Control Logic Call Reneging.doe**: Example model of Call Reneging at a Call Center.

-**Control Logic Other Type Schedule.doe**: Illustrates the use of an *Other* type schedule.

-**Resources Using Statesets.doe**: Illustrates the use of a user defined state and stateset to collect resource frequency statistics.

-**Flow Process Reduce Regulator Flow Rate as Tank Drains.doe**: Illustrates how to reduce a regulator's flow rate based on a tank level.

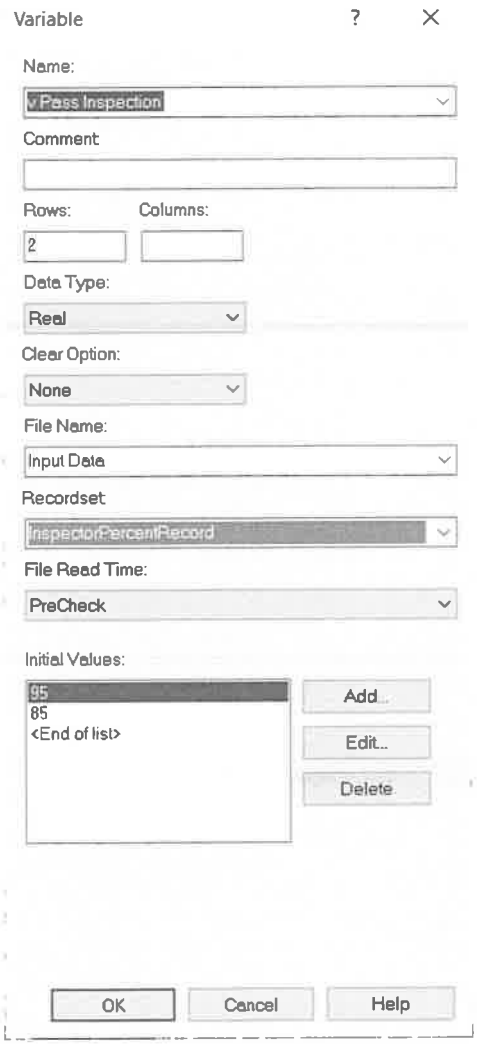
Addendum

Alternative (preferred) method of reading items from an Excel spreadsheet.

(To be read in conjunction with pages 182 & 187)

Read the Passing Inspection Rates – Modify the Variable spreadsheet to read from Excel file

Add in the File Name, Recordset and File Read Time fields to fill this variable with values read from the Excel spreadsheet file prior to the Arena run.



Now continue in the main text on page 187 (Examine Inspection Pcts File)