



*Università degli Studi di
Padova*

*Facoltà di
Ingegneria*

*Dipartimento di Tecnica e Gestione dei Sistemi
Industriali*

**COMPUTER SIMULATION
AND
ARENA SOFTWARE**

Relatore: Ch.mo Prof. Giorgio Romanin Jacur

Laureando: Iacopo Mazzi

**Anno Accademico
2010/2011**

TABLE OF CONTENTS

SUMMARY	5
CHAPTER 1: INTRODUCTION TO SIMULATION	6
<u>1.1 INTRODUCTION.....</u>	6
<u>1.2 COMPUTER SIMULATION</u>	7
1.2.1 PROS AND CONS OF COMPUTER SIMULATION	7
1.2.2 DIFFERENT KINDS OF SIMULATION	7
CHAPTER 2: QUEUEING THEORY	8
<u>2.1 BASES OF QUEUEING MODELS.....</u>	8
<u>2.2 LABELLING OF QUEUEING MODELS</u>	9
2.2.1 FORMAL MODEL OF M TYPE ARRIVALS	9
2.2.2 FORMAL MODEL OF M TYPE SERVICE	10
<u>2.3 M/M/S QUEUES</u>	10
CHAPTER 3: FUNDAMENTAL SIMULATION CONCEPTS	11
<u>3.1 PROBABILITY DISTRIBUTIONS</u>	11
3.1.1 DISCRETE DISTRIBUTION	12
3.1.2 ERLANG DISTRIBUTION	12
3.1.3 EXPONENTIAL DISTRIBUTION	13
3.1.4 NORMAL DISTRIBUTION	13
3.1.5 TRIANGULAR DISTRIBUTION	14
3.1.6 UNIFORM DISTRIBUTION	14
3.1.7 WEIBULL DISTRIBUTION	15
<u>3.2 PARTS OF A SIMULATION MODEL</u>	16
3.2.1 ENTITIES	16
3.2.2 ATTRIBUTES	16
3.2.3 VARIABLES	16

3.2.4 RESOURCES	17
3.2.5 QUEUES	17
3.2.6 EVENTS	17
3.2.7 SIMULATION CLOCK	18
3.2.8 STATISTICAL ACCUMULATORS	18
<u>3.3 OVERVIEW OF A SIMULATION STUDY</u>	18
<u>3.4 BUILDING A SIMPLE MODEL</u>	19
3.4.1 THE CREATE MODULE	20
3.4.2 THE ENTITY DATA MODULE	20
3.4.3 THE PROCESS FLOWCHART MODULE	21
3.4.4 THE DISPOSE MODULE	21
3.4.5 DYNAMIC PLOTS	21
3.4.6 RESOURCE ANIMATION	23
CHAPTER 4: A BASIC MODEL	25
<u>4.1 AN ELECTRONIC ASSEMBLY AND TEST SYSTEM</u>	25
4.1.1 SYSTEM DESCRIPTION	25
4.1.2 BUILDING THE MODEL	26
4.1.3 RUNNING THE MODEL	32
4.1.4 VIEWING THE RESULTS	34
<u>4.2 THE ENHANCED ELECTRONIC ASSEMBLY AND TEST SYSTEM</u>	36
4.2.1 SYSTEM DESCRIPTION	36
4.2.2 SCHEDULES AND STATES	36
4.2.3 RESOURCE FAILURES	38
4.2.4 FREQUENCIES	39
4.2.5 RESULTS	40
CHAPTER 5: A REALISTIC MODEL	41
<u>5.1 SYSTEM DESCRIPTION</u>	41
<u>5.2 MODELING APPROACH</u>	43

<u>5.3 BUILDING THE MODEL</u>	44
5.3.1 DATA MODULES	44
5.3.2 LOGIC MODULES	47
<u>5.4 SYSTEM ANIMATION</u>	53
<u>5.5 SYSTEM OVERVIEW</u>	55
CONCLUSION	57
REFERENCES	59
ACKNOWLEDGEMENTS	60

SUMMARY

In modern world simulation has become a fundamental tool in analyzing real life situations. Creating a model can sometimes be the only way to study a project and most of times it is the most fruitful method.

Through simulation it is possible to manage bottlenecks, improve order-management systems and to estimate warehouse capacity testing different plausible scenarios to improve the process before running the real system.

The aim of this work is first explain basic concepts of simulation, and in particular computer simulation, and then apply them to concrete situations, from the simplest to the more realistic ones.

The expected result is to give the reader a basic knowledge of how simulation can be helpful in different occasions and to provide the means to create basic and intermediate realistic models.

CHAPTER 1: Introduction to simulation

1.1.Introduction

Simulation attempts to model real-life or hypothetical situations to study how the system works, usually with appropriate softwares.

Computer simulation deals with models of systems. A system can be defined as a process, either actual or planned.

Examples of different systems can be:

- A distribution network of plants, warehouses and transportation links;
- A supermarket with inventory control, checkout, and customer service;
- A manufacturing plant with machines, people, transport devices, conveyor belts and storage space.

Sometimes it might be possible to experiment with the actual physical system, like experimenting with different traffic lights sequences, but sometimes it is too difficult, costly or impossible to do physical studies on the system itself like in the case of an existing factory (it might be extremely costly to change to an experimental layout).

Two of the most important kinds of models that can be highlighted are: physical models and logical (or mathematical) models:

- Physical models can be both tabletop models that are miniature versions of the facility and a full-scale version simulating the facility itself.
- Logical (or mathematical) models are usually represented in a computer program that is exercised to address questions about the model's behavior.

If the model is simple enough it is possible to use traditional mathematical tools like queueing theory, differential-equation methods or linear programming but most systems subject of study are so complicated and may not be possible to find exact mathematical solutions, which is where simulation comes in.

In this discussion will be analyzed only logical models using a simulation software, Arena[®] by Rockwell Automation.

1.2.Computer simulation

1.2.1.Pros and cons of computer simulation

Computer simulation refers to methods for studying a wide variety of models reproducing a wide variety of real-world systems by numerical evaluation; in the last three decades simulation has been considered as the most powerful and popular operations research tool. The reason for this is that the model can be allowed to become extremely complex, so it can represent faithfully the system while other methods may require stronger simplifying assumptions about the system to enable analysis, which might bring the validity of the model into question.

On the other hand simulation is not always as easy as it seems: many real systems are affected by random inputs, and consequently many simulation models involve stochastic input components causing their output to be random too.

1.2.2.Different kinds of simulation

There are many different ways to classify simulation models, but a useful way is along these three dimensions:

- **Dynamic or Static:** Time plays a natural role in static models but does not in dynamic ones. Most operational models are dynamics and Arena was designed to best fit with this kind of models.
- **Discrete or Continuous:** In a discrete model changes occur only at specified points in time while in a continuous model the state of the system changes continuously over time. A discrete model can be a manufacturing system where parts arrive and leave following a specific timetable; a water reservoir with water flowing in and out is a perfect example of a continuous model. In the same model can be present elements of both discrete and continuous change: these models are called mixed continuous-discrete models.
- **Deterministic or Stochastic:** Models with no random inputs are called deterministic models while stochastic models operate with at least some random inputs. Due to the randomness of the inputs, even the outputs of a stochastic model are uncertain and the analyst has to consider this carefully in designing and interpreting the results of this kind of projects.

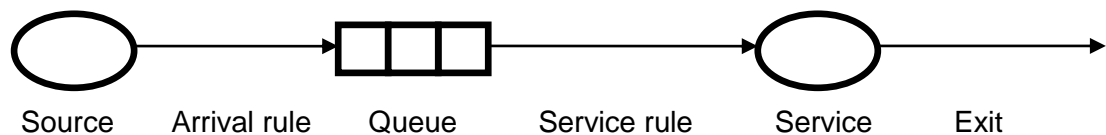
CHAPTER 2: Queueing theory

2.1. Bases of queueing models

Queues (or waiting lines) are necessary when customers wait for a service as the service capacity is not sufficient to supply the service at once.

In a queueing model:

- Customers are generated by an input source.
- Customers arrive according to an arrival rule.
- Customers enter the system by joining a queue.
- A customer is chosen according to a queue discipline.
- The chosen customer obtains the required service.
- The service follows a service rule.
- The customer who obtained the service exits the system.



The input source may have either infinite or finite population; if the population is infinite it is not altered by additional customers entering the system.

Customer arrival is ruled by the interarrival time distribution.

Queues may be either infinite or finite (in this case we cannot exceed a maximum capacity).

Queue discipline may be FIFO (First In First Out), LIFO (Last In Last Out), SIRO (Service In Random Order), sorted (according to a priority), PS (Processor Shared) or AS (Ample Server).

It is possible to have one or multiple servers; if multiple we have parallel service channels.

2.2. Labelling of queueing models

Queueing models can be represented using Kendall's notation: $A/B/S/K/N/D$. "A" is the interarrival time distribution (or arrival rule), "B" is the service time distribution, "S" is the number of servers (parallel channels), "K" is the maximum system capacity or the maximum number of customers in the system, "N" is the source population and "D" is the service discipline assumed (FIFO, LIFO, SIRO, PRIOR, PS or AS).

Some standard notation for interarrival time and service time distributions are: M for a memoryless or Markovian (exponential) time, G for general (arbitrary) time, D for degenerate distribution and E_k for an Erlang distribution with k phases.

By default the service discipline is considered as First In First Out (FIFO), while both system capacity and source population are assumed as infinite so, using Kendall's notation, we can have for example $M/M/1$ or $M/G/1$ or $M/M/S$.

2.2.1. Formal model of M type arrivals

Given 0 as current instant, t as any instant > 0 and τ as the instant of next arrival, the process may be described in two equivalent ways:

$$\text{Prob}(\tau \in [t, t + \Delta t] \mid \tau > t) = \lambda \Delta t + o(\Delta t)$$

$$\text{Prob}(\tau > t) = e^{-\lambda t}$$

Where λ is a positive real number and $o(\Delta t)$ is an infinitesimal of upper order with respect to Δt .

Obviously the interarrival time is exponentially distributed, λ is the arrival rate, or the mean number of arrivals in time unit and $1/\lambda$ is the mean interarrival time.

The value of λ may be a constant (infinite source population and homogeneous system with infinite capacity), dependent on the number of customers in the system (finite source population or finite capacity of the system) or dependent on other parameters.

This kind of process is considered without memory, or rather the evolution newly begins at every instant.

The sum of l markovian processes with $\lambda_1, \lambda_2, \dots, \lambda_l$ is a markovian process with $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_l$.

2.2.2. Formal model of M type service

Given 0 as current instant, t as any instant > 0 and τ as the instant of service end, even M type service may be described in two equivalent ways:

$$\text{Prob}(\tau \in [t, t + \Delta t] \mid \tau > t) = \mu \Delta t + o(\Delta t)$$

$$\text{Prob}(\tau > t) = e^{-\mu t}$$

Where μ is a positive real number and $o(\Delta t)$ is an infinitesimal of upper order with respect to Δt .

In this kind of service μ is the mean number of services in time unit, or the service rate, if all servers are always busy, $1/\mu$ is the mean service time and the service time is exponentially distributed.

The service rate may be constant (only one server and any discipline), dependent on the number of customers in the system (multiple servers) or dependent on other parameters.

In this case, as well as M type arrivals, the process is considered without memory.

2.3.M/M/S queues

Considering n_0 as the initial state, if $n_0=1, 2, \dots, N$ then the queue will remain in the same state for a time τ , exponentially distributed with parameter $\lambda+\mu$, after which there will occur: either an arrival with probability $\lambda/(\lambda+\mu)$ or an exit with probability $\mu/(\lambda+\mu)$. If $n_0=0$ then the queue will remain in the same state for a time τ , exponentially distributed with parameter λ , after which there will occur an arrival with probability 1.

After the transition from the initial state the queue will evolve in the same way from the state just reached, without any memory of its previous evolution.

CHAPTER 3: Fundamental Simulation Concepts

3.1. Probability distributions

Arena software contains various built-in functions for generating random variates from the commonly used probability distributions. Each distribution has in Arena one or more parameter values associated with it and is required to specify these parameter values. A summary of the distributions and parameter values is given in Table 3-1.

Distribution		Parameter Values	
Beta	BETA	BE	Beta, Alpha
Continuous	CONT	CP	CumP ₁ , Val ₁ ,...CumP _n , Val _n
Discrete	DISC	DP	CumP ₁ , Val ₁ ,...CumP _n , Val _n
Erlang	ERLA	ER	ExpoMean, k
Exponential	EXPO	EX	Mean
Gamma	GAMM	GA	Beta, Alpha
Johnson	JOHN	JO	Gamma, Delta, Lambda, Xi
Lognormal	LOGN	RL	LogMean, LogStd
Normal	NORM	RN	Mean, StdDev
Poisson	POIS	PO	Mean
Triangular	TRIA	TR	Min, Mode, Max
Uniform	UNIF	UN	Min, Max
Weibull	WEIB	WE	Beta, Alpha

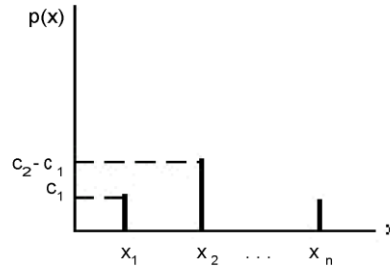
Table 3-1. Arena's probability distributions

These distributions can be specified by using two different formats: the primary format can be selected by using the variable's full name or a four-letter abbreviation of the name (first or second column of Table 3-1), the secondary format is selected by specifying the distribution with a two-letter abbreviation (third column of the table).

In the primary format it is necessary to explicitly enter the parameters of the distribution as arguments, for example TRIA(1, 2, 3); in the secondary format the parameters refer to a parameter set within the Elements panel, for example TR(ProcessTime) specifies a Triangular distribution with the min, mode and max defined in the ProcessTime parameter set.

3.1.1. Discrete distribution

DISCRETE(CumP₁, Val₁,..., CumP_n, Val_n) or DISC(CumP₁, Val₁,..., CumP_n, Val_n) or DP(ParamSet)



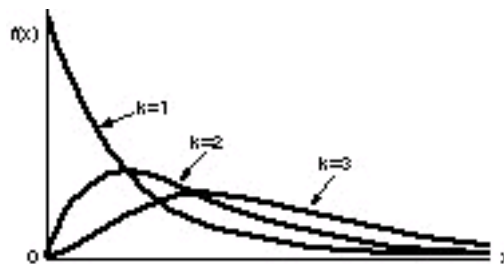
This discrete empirical distribution is used to incorporate discrete empirical data directly into the model. It is frequently used for discrete assignments such as job type, the visitation sequence or the batch size for an arriving entity.

The Discrete function returns a sample from a user-defined discrete probability distribution defined by the set of n possible discrete values (x₁,...,x_n) that can be returned by the function and the cumulative probabilities (c₁,...,c_n) associated with these discrete values.

$$p(x_j) = c_j - c_{j-1} \quad \text{where } c_0 = 0$$

3.1.2. Erlang distribution

ERLANG(ExpMean, k) or ERLA(ExpMean, k) or ER(ParamSet)



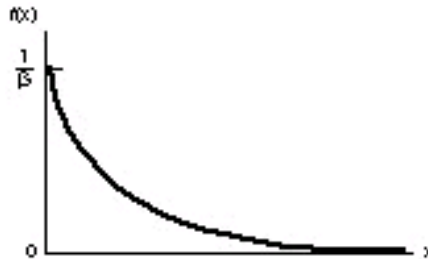
The Erlang distribution is commonly used when an activity occurs in successive phases and each of these has an exponential distribution; This distribution is often used to represent the time required to complete a task.

The mean (β) of each of the component exponential distributions and the number of exponential random variables (k) are the parameters of this distribution.

$$f(x) = \begin{cases} \frac{\beta^{-k} x^{k-1} e^{-x/\beta}}{(k-1)!} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

3.1.3.Exponential distribution

EXPONENTIAL(Mean) or EXPO(Mean) or EX(ParamSet)



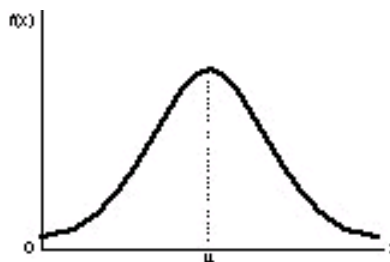
The exponential distribution is used to model intervent times in random arrival and break-down processes, but is usually inappropriate for modeling process delay times.

The mean (β) is the parameter of this distribution, and is a positive real number.

$$f(x) = \begin{cases} \frac{1}{\beta} e^{-x/\beta} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

3.1.4.Normal distribution

NORMAL(Mean, StdDev) or NORM(Mean, StdDev) or RN(ParamSet)



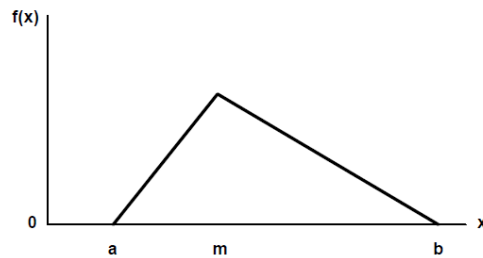
This kind of distribution is used when the central limit theorem applies and, empirically, for processes that appear to have a symmetric distribution. Because normal distribution has a theoretical range of $(-\infty, +\infty)$ it can not be used for positive quantities like processing times

The mean (μ) is a real number and standard deviation (σ) is a positive real number.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \quad \text{for all real } x$$

3.1.5.Triangular distribution

TRIANGULAR(Min, Mode, Max) or TRIA(Min, Mode, Max) or TR(ParamSet)



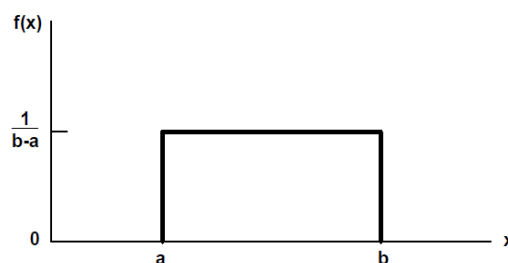
This distribution is used when the exact form of the distribution is unknown but are available estimates for the minimum, the most likely and the maximum values.

The minimum (a), mode (m) and maximum (b) values for the triangular distribution are real numbers with $a < m < b$.

$$f(x) = \begin{cases} \frac{2(x-a)}{(m-a)(b-a)} & \text{for } a \leq x \leq m \\ \frac{2(b-x)}{(b-m)(b-a)} & \text{for } m \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

3.1.6.Uniform Distribution

UNIFORM(Min, Max) or UNIF(Min, Max) or UN(ParamSet)



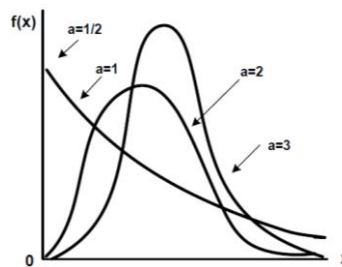
The uniform distribution is used when all values over a finite range are considered to be equally likely; it is sometimes used when no information other than the range is available.

The minimum (a) and the maximum (b) values are real numbers with $a < b$.

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases}$$

3.1.7. Weibull distribution

WEIBULL(Beta, Alpha) or WEIB(Beta, Alpha) or WE(ParamSet)



The weibull distribution is used in reliability models to represent the lifetime of a device. If a system consists of a large number of parts that fail independently, and if this system fails when any single part fails, then the time between successive failures follows the weibull distribution.

Scale parameter (β) and shape parameter (α) are positive real numbers.

$$f(x) = \begin{cases} \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

3.2.Parts of a simulation model

3.2.1.Entities

Many simulation, or more properly most of them, involve entities that move around the system, affect and are affected by other entities and the state of the system, change status and affect, obviously, the output performance measures. Entities are the dynamic objects involved in Arena simulation. All entities have to be created, either by the analyst or automatically by the software. Usually, while modeling a system, entities are the first thing to think about to have an idea of what have to be represented in the system itself. Most entities represent “real” things in a simulation but sometimes could be useful to use logic entities, which do not correspond to anything tangible but can be conjured up to take care of certain modeling operations.

3.2.2.Attributes

Attributes are common characteristics of all entities, with a specific value that can differ from one entity to another and are attached to Arena entities to individualize them.

Their values are tied to specific entities: the same attribute will usually have different values for different entities

As an instance part entities could have different attributes called Priority, Color and Due Date to indicate these characteristic for each individual entity; is up to the analyst to figure out what attributes an entity needs, name them, assign values to them, change them as appropriate and then use them when it is time.

3.2.3.Variables

A variable (or global variable) is a piece of information that reflects some characteristic of the system, regardless of what kinds or how many entities might be in the system itself.

There are two different kinds of variables: user-defined variables or Arena built-in variables. For instance user-defined variables should be mean service time, travel time or current shift while Arena built-in variables should be number in queue, current simulation clock time or number of busy servers.

Unlike attributes, variables are not tied to any specific entity, but rather pertain to a system at large.

As an example the time to move between two subsequent stations in a model might be the same irrespective of the stations, so a variable called TransferTime could be defined and set to the appropriate value and used whenever this constant time is needed.

3.2.4.Resources

Resources in arena represents things like personnel, equipment or space in a storage area of limited size. An entity seizes units of a resource when available and releases it (or them) when finished.

A resource can represent a group of several individual servers, each of which is called a unit of the resource.

3.2.5.Queues

When an entity cannot move on, perhaps because it needs to seize a unit of a resource that is tied up by another entity, it needs a queue to wait for that resource.

In Arena queues can have limited capacities to represent, for example, limited buffer space.

3.2.6.Events

An event is something that happens at a certain instant of simulated time that might change attributes, variables or statistical accumulators.

To execute a simulation has to keep track of the events that are supposed to happen in the simulated future: this information is stored in an event calendar.

In our first example there are three kinds of events:

- Arrival: A new part enters the system.
- Departure: A part finished its service at the drill press and leaves the system.
- The End: The simulation is stopped after 20 minutes.

3.2.7.Simulation Clock

The current value of time in the simulation is simply held in a variable called the simulation clock. Unlike real time the simulation clock does not take on all values and flow continuously but it lurches from the time of one event to the time of the next event scheduled to happen.

While we will keep track of the simulation clock and event calendar ourselves in the hand simulations, these are clearly important pieces of any dynamic simulation so Arena keeps track of them: the clock is a variable called TNOW.

3.2.8.Statistical Accumulators

To get output performance measures the analyst has to keep track of various intermediate statistical accumulator variables such as the number of parts produced so far, the longest time spent in queue, the longest time in system seen so far, and so on.

3.3.Overview of a simulation study

Deciding how to model a system several aspects tend to come up frequently:

- Understand the system: The analyst must have an intuitive, down-to-earth feel of how this particular system works.
- Be clear about goals: Realism has to be the watchword; one of the first things to do is understand what can be learned from the study and specify about what is to be observed, manipulated, changed and delivered.
- Translate into modeling software: Once the modeling assumptions are agreed upon, represent them faithfully in the simulation software.
- Verify that your computer representation represents the conceptual model faithfully: Probe the extreme regions of the input parameters, verify that the right things happen with obvious input.
- Validate the model: Do the input distributions match what has been observed in the real system? Do the output performance measures from the model match up with those from reality?
- Analyze the results: It is fundamental to carry out the right kinds of statistical analyses to be able to make accurate and precise statements.

3.4. Building a simple model

Introducing Arena software we start from a very simple case of a model representing queues or waiting lines.

In a portion of a manufacturing facility, “blank” parts arrive to the drilling center; if the drill press is idle the parts is processed instantly, otherwise it waits in a FIFO (First In First Out) queue. After the drill process the parts leave the system. In Figure 3-1 is possible to see the logical structure of this first simple system.

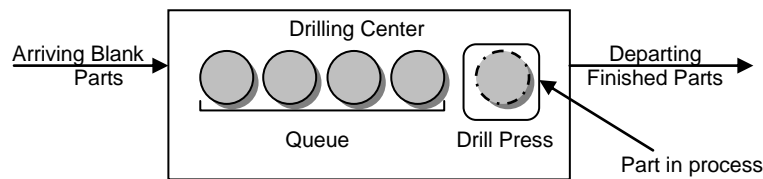


Figure 3-1. Drilling system

First of all open Arena software and attach the panels needed if they are not in the Project Bar (on the left side of the screen). To attach a panel, identifiable by its .tpo file extension, just click on File > Template Panel > Attach and choose the desired panel from the Template folder, usually this folder is under the Arena folder, as shown in Figure 3-2. To create this simple model we need only the Basic Process Panel.

This model requires one instance of each of three flowchart modules: Create, Process and Dispose; to add an instance just drag its icon from the Project Bar to the flowchart view of the model and drop it.

If Object > Auto Connect is selected and the modules are dragged in the sequential order of the model Arena will connect them in the right order (as shown in Figure 3-3).

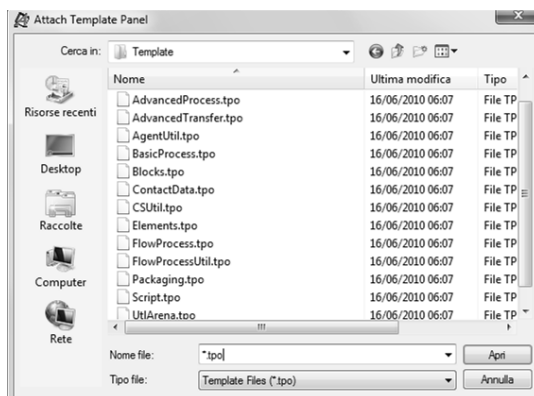


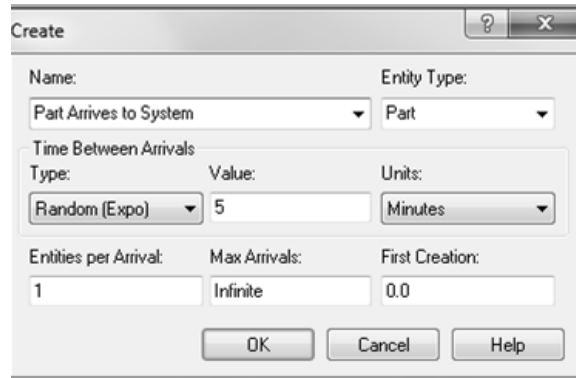
Figure 3-2. The Attach Template Pan



Figure 3-3. Initial Placement

3.4.1.The Create Module

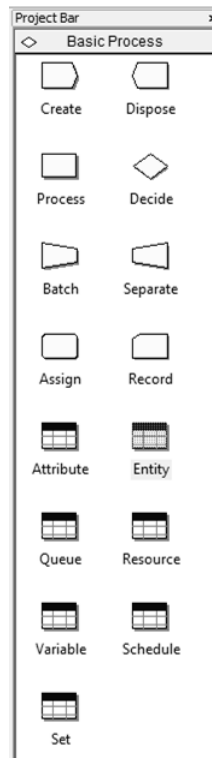
Open the Create module by double clicking it: edit the Name of the instance to Part Arrives to System, the Entity type to Part, the Time Between Arrivals Value to 5 and Time Units to Minutes. The complete dialog box is shown in Display 3-1.



Display 3-1. Part Arrives to System

3.4.2.The Entity Data Module

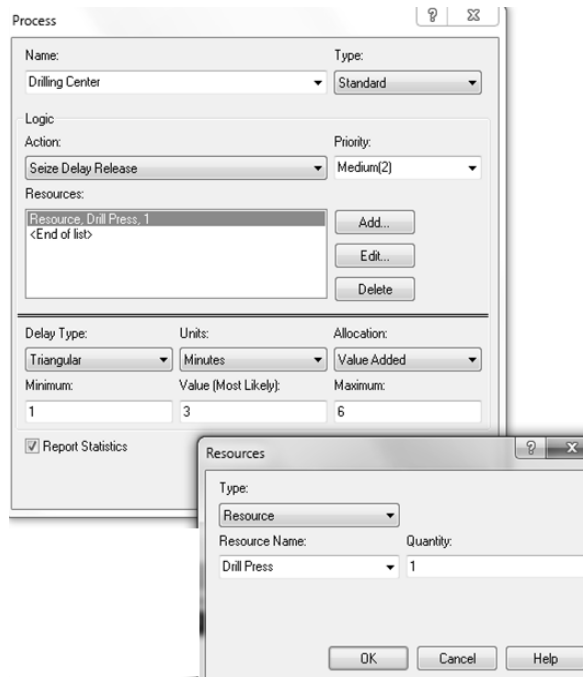
Clicking the Entity data module in the Project Bar (Display 3-2) is possible to see all the entities involved in this simulation. In this first example the only entity is Part and we want to change its picture from “Report” to “Blue Ball”: to do this just choose Picture.Blue Ball from the Initial Picture List.



Display 3-2. Project Bar

3.4.3.The Process Module

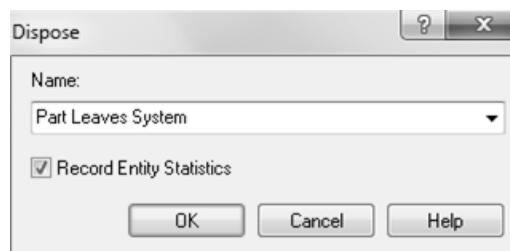
As shown in Display 3-3 since the Action specified includes Seizing a Resource, it is necessary to click the Add button and open the Drill Press Resource in the secondary dialog box.



Display 3-3. The Process Module

3.4.4.The Dispose Module

The only edited thing is the default name which has been changed to Part Leaves System, as shown in Display 3-4.



Display 3-4. The Dispose Module

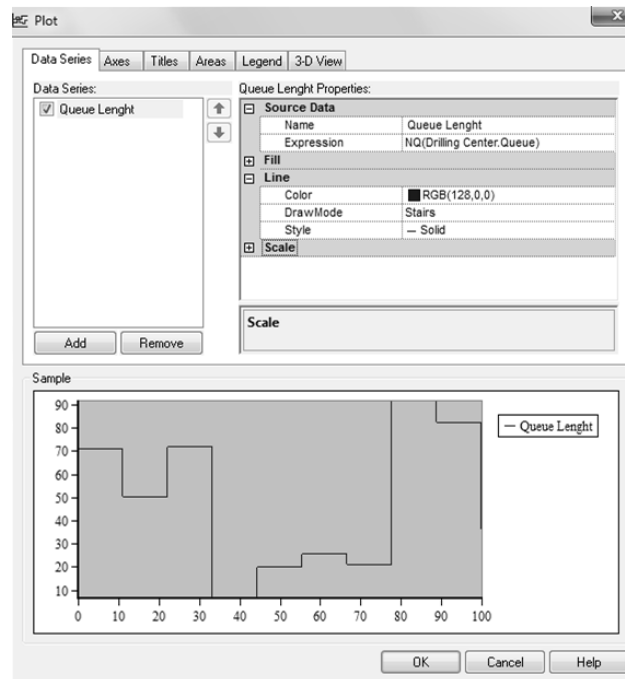
3.4.5.Dynamic Plots

To make a plot from a scratch just press the Plot Button on the Animate toolbar (Display 3-5) to get a blank Plot dialog box then edit data while watching the sample area at the bottom to preview the plot setup, though the curve is just a schematic and will not match the data generated during the simulation.



Display 3-5. The Animate Toolbar

In the Data Series tab we change the name to Queue Length, the expression to NQ(Drilling Center.Queue) and the Line Style to Stairs as shown in Display 3-6. The expression NQ(Drilling Center.Queue) shows the current number in queue and can be written in the Expression field of the Data Series tab or chosen from the Expression Builder accessible via the pull-down in the Expression field (Display 3-7).



Display 3-6. Plot Dialog Box

In the Axes tab > Time (X) Axis we modify the Scale > Maximum to 20, Scale > MajorIncrement to 5, Title > Text to Time (Minutes) and put true to Title > Visible. In the Left Value (Y) Axis Properties we change Scale > MajorIncrement to 1, Title > Text to Queue Length and choose True on Title > Visible.

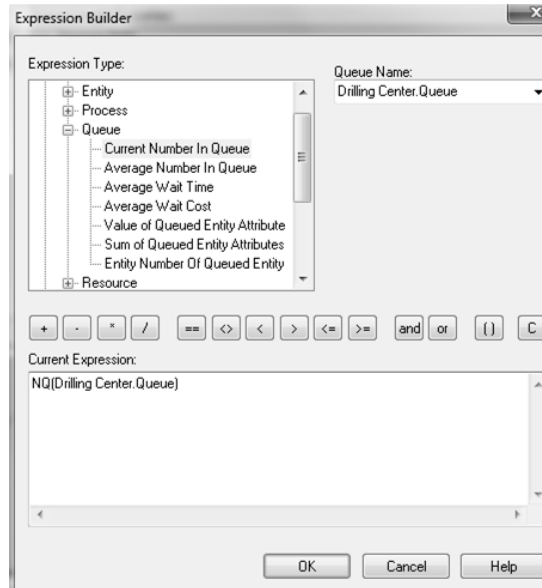
In the Legend tab just uncheck the Show Legend box.

The plot for the number busy at the drill press is quite similar so duplicate the Queue Length one (Ctrl+D) and edit it.

In the Data Series tab > Source Data change the name to Number Bust and the Expression to NR(Drill Press); this expression can be found in the Expression Builder:

Basic Process Variables → Resource → Usage → Current Number Busy and select Drill Press under Resource Name.

In the Axes tab > Left Value(Y) Axis under Scale change the Maximum to 2 and set AutoScaleMaximum to False from the pull-down; under Title set Text to Number Busy.



Display 3-7. Expression Builder

3.4.6.Resource Animation

Animating the state of a Resource (showing when the Drill Press is Idle or Busy in our first example) can be a good idea to give a visual idea of how the system works.

Search on Tool > Arena Symbol Factory the best picture that represents the Resource, for the Drill Press I chose the pictures shown in Display 3-8 and Display 3-9 for the Idle and the Busy state.



Display 3-8. Press Idle

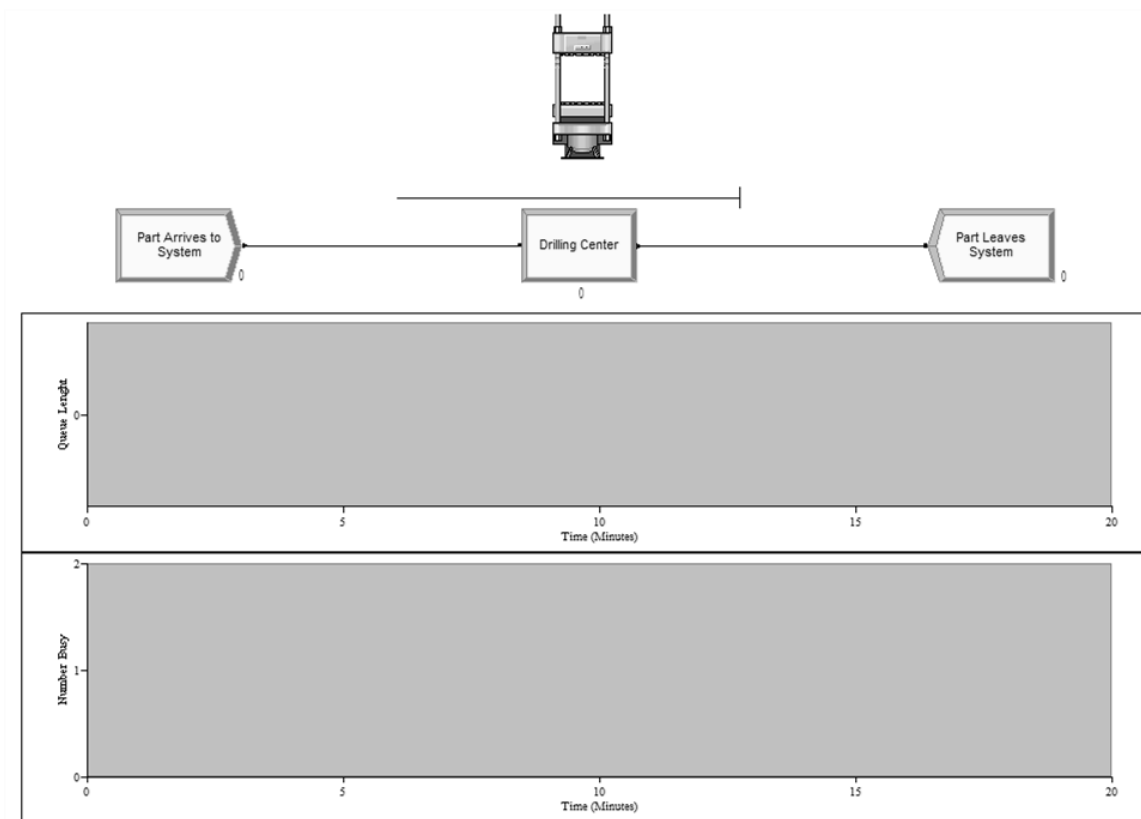


Display 3-9. Press Busy

Copy the selected image, open the Resource button on the Animate toolbar and from the identifier pull-down select the desired resource (in our situation the only resource is Drill Press). Delete the Inactive and Failed status from the left list because they are not

present in this simulation. In the right list click add, open the blank icon, paste the previously copied image and close the window, this image will appear in this pictures set. Repeat the procedure for the other picture. To associate the picture to the status select the Idle status from the left list, the picture shown in Display 3-8 from the right list and press « (the same has to be done for the Busy status and the picture shown in Display 3-9). Save the library and then close the window: now select the area where the picture has to appear.

The completed model should appear as in Display 3-10.



Display 3-10. The Completed Model

CHAPTER 4: A BASIC MODEL

4.1. An Electronic Assembly and Test System

4.1.1. System description

This first system represents the final operations of the production of two different sealed electronic units, as shown in Figure 4-1

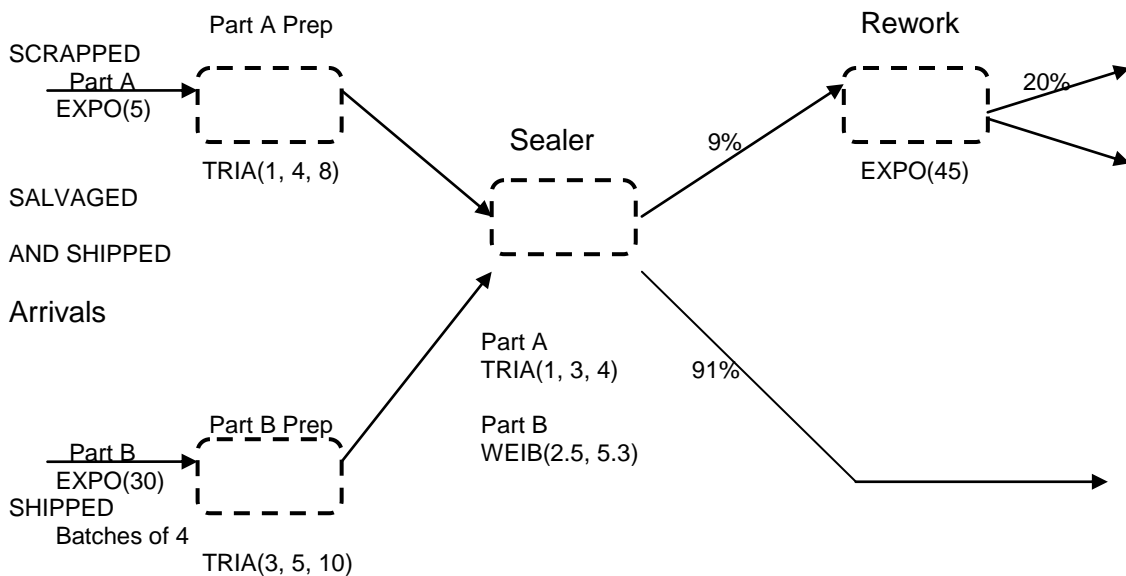


Figure 4-1. Plan of the System

The first units, called Part A, are produced in an adjacent department, outside the bounds of this model, with interarrival times to our model being exponentially distributed with a mean of 5 (all times are expressed in minutes).

Upon arrival, they are transferred instantly to the Part A Prep area, where the part is deburred and cleaned; the process time for the combined operation at the Part A Prep area follows a TRIA(1, 4, 8) distribution. The part is then transferred instantly to the sealer.

The second units, called Part B, are produced in a different building, also outside this model's bounds, where they are held until a batch of four units is available; the batch is then sent to the final production area, object of this model. The time between the arrivals of successive batches of Part B to our model is exponential with a mean of 30 minutes. Upon arrival at the Part B Prep area, the batch is separated into the four individual units, which are processed individually from here on, and the individual parts proceed (instantly) to the Part B Prep area. The processing at the Part B Prep area has

the same two steps as at the Part A Prep area, except that the process time for the combined operations follows a TRIA(3, 5, 10) distribution. The part is then sent instantly to the sealer.

At the sealer operation, the electronic components are inserted, the case is assembled and sealed, and the sealed unit is tested. The total process time for these operations depends on the part type: TRIA(1, 3, 4) for Part A and WEIB(2.5, 5.3) for Part B (where 2.5 is the scale parameter β and 5.3 is the shape parameter α). Ninety-one percent of the parts pass the inspection and are transferred immediately to the shipping department; whether a part passes is independent of whether any other parts pass.

The remaining parts are transferred instantly to the rework area where they are disassembled, repaired, cleaned, assembled and re-tested. Eighty percent of the parts processed at the rework area are salvaged and transferred instantly to the shipping department as reworked parts, the rest are transferred instantly to the scrap area. The time to rework a part follows an exponential distribution with mean of 45 minutes and is independent of part type and the ultimate disposition (salvaged or scrapped).

The purpose of this simulation is to collect statistics in each area on resource utilization, number in queue, time in queue and the cycle time (or total time in system) separated out by shipped parts, salvaged parts or scrapped parts.

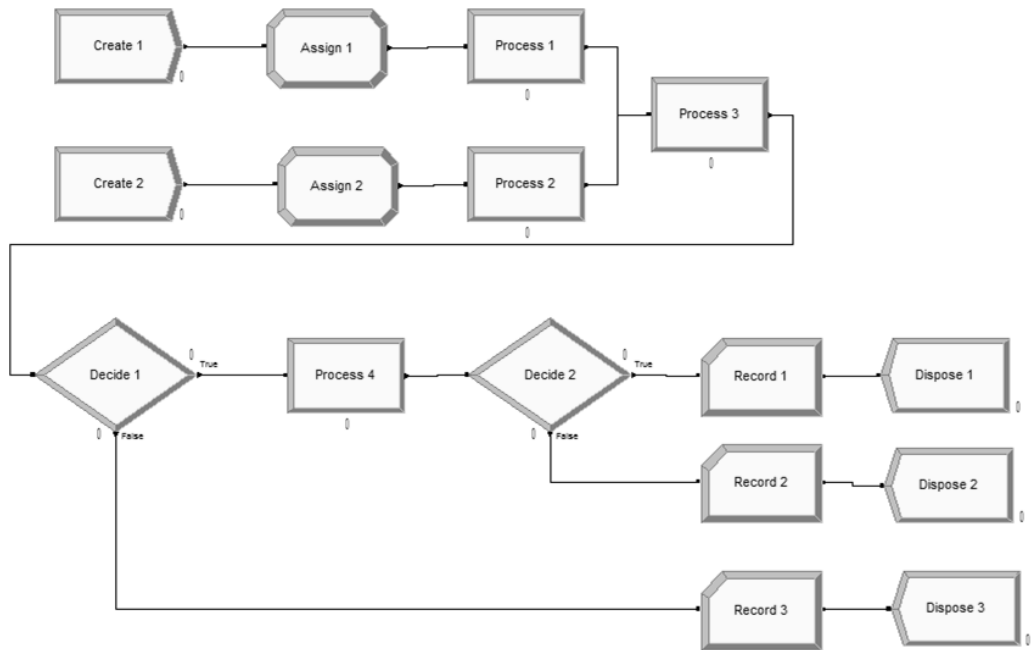
Initially the simulation will be run for four consecutive 8-hour shifts, or 1920 minutes.

4.1.2. Building the model

First of all open a new window and place the necessary modules on the screen. The model we are going to develop is constituted by two Create modules, two Assign, four Process, two Decide, three Record and three Dispose modules.

After having disposed these models as shown in Display 4-1 connected by the Auto-Connect feature, or the Connect feature which allows the user to connect as he prefers (both in the Object menu), we have to enter the required information in each module.

Part A Arrive: (Display 4-2) we changed the name to Part A Arrive, and specified as Part A the Entity Type; the Time Between Arrivals is set as Random with a Value of 5 Minutes (following an exponential distribution with a mean of 5 minutes). The remaining entries are the default options.



Display 4-1. Modules Disposition

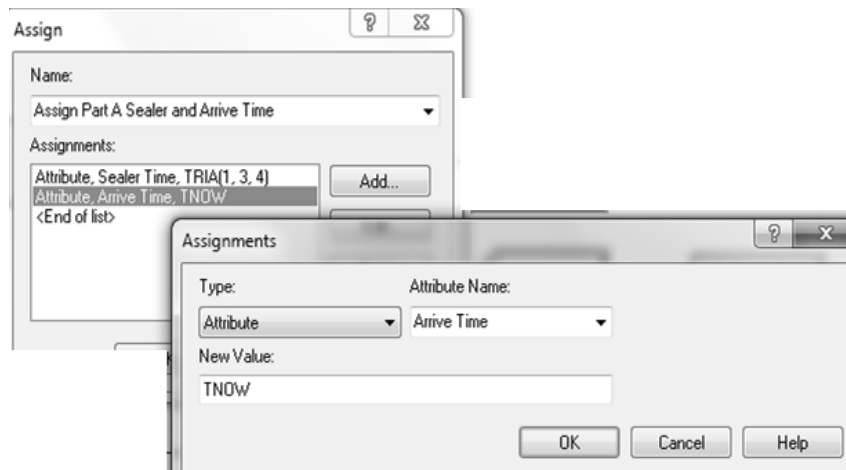
Part B Arrive: (Display 4-3) it is very similar to that for Part A, except for the interarrival time (30 minutes instead of 5) and the number of Entities per Arrival reflecting the batches of four.

Display 4-2. Part A Arrive

Display 4-3. Part B Arrive

Assign Part A Sealer and Arrive Time: (Display 4-4) after having created the arriving parts it is necessary to define a Sealer Time attribute and assign it the processing time following the TRIA(1, 3, 4) distribution. We add the Arrive Time attribute to record the arrival time of the entity; its value is the variable TNOW which provides the current simulation time, in this case the time the part arrived or was created.

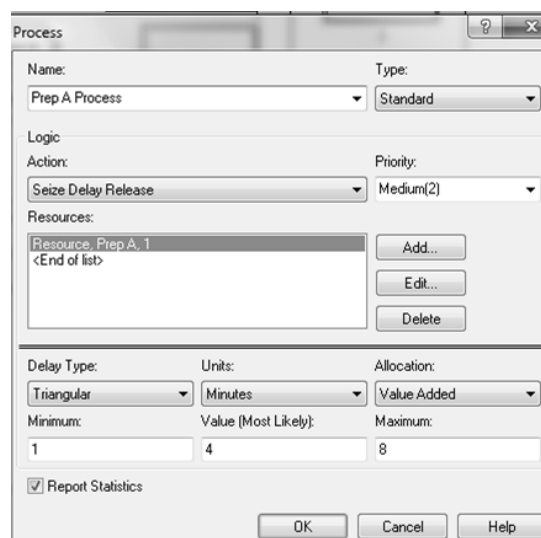
Assign Part B Sealer and Arrive Time: here we have to assign the same information put in the Assign Part A module, obviously changing the sealer time (here following the weibull distribution WEIB).



Display 4-4. Assign Part A Sealer and Arrive Time

Prep A Process: (Display 4-5) the Process module has four possible Actions:

- Delay: it will cause an entity to undergo a specified time delay; this action does not require a Resource. This implies that waiting will occur and that multiple entities could undergo the Delay simultaneously.
- Seize Delay: it provides the waiting and delaying, but it does not release the Resource at the end of processing to be available for the next entity. If this Action is used, it is assumed that the Resource would be Released downstream in another module.
- Seize Delay Release: this action is similar to Seize Delay, but entities release Resource units after Delay.
- Delay Release: assumes that the entity previously Seized a Resource and will undergo a Delay here, followed by the Release for the Resource.

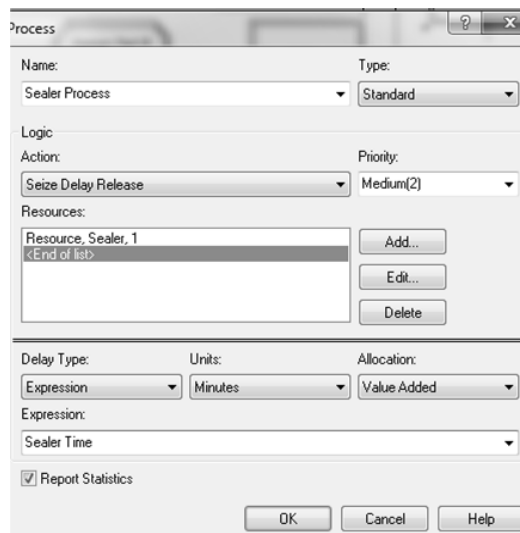


Display 4-5. Prep A Process

This module requires a Seize Delay Release Action, so is necessary to add a Resource called Prep A. considering that this area follows a TRIA distribution, we choose Triangular Delay Type (with minutes as time unit) and 1, 4, 8 as the minimum, most likely and maximum value.

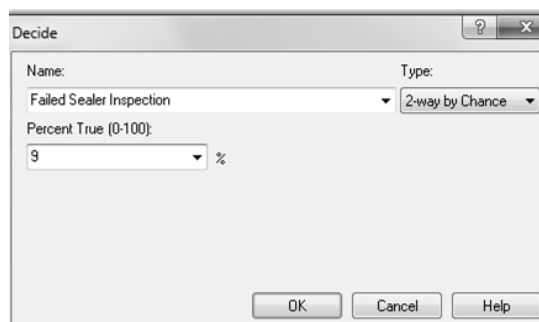
Prep B Process: it is filled out in an almost-identical fashion, with the exception of the name, the resource name and the process time parameters. In this module the resource name is Prep B, obviously, and the values are 3, 5, 10.

Sealer Process: (Display 4-6) in this module we simulate the sealer activity. Note that the Sealer Time was defined in the previous modules, so we can use it as an Expression. When an entity seizes this resource it will undergo a process delay equal to the value contained in its sealer time attribute.



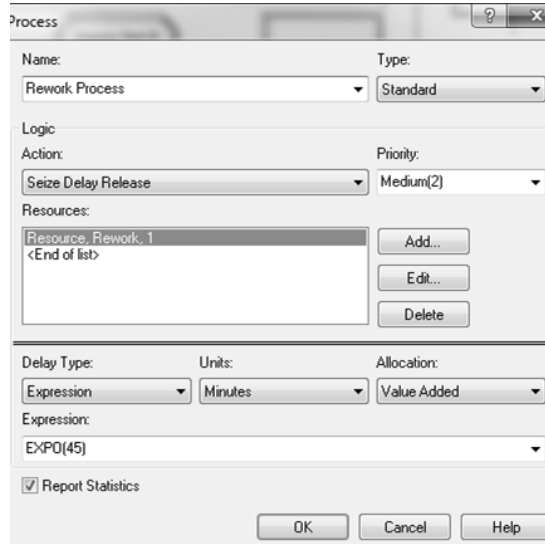
Display 4-6. Sealer Process

Failed Sealer Inspection: (Display 4-7) this Decide module simulates the inspection following the sealer operation with a 2-way by Chance Type. In this model the Percent True (9) represents the amount of entities failing the inspection; 9% of entities follow the True branch and 91% (the passed items) follow the False branch.



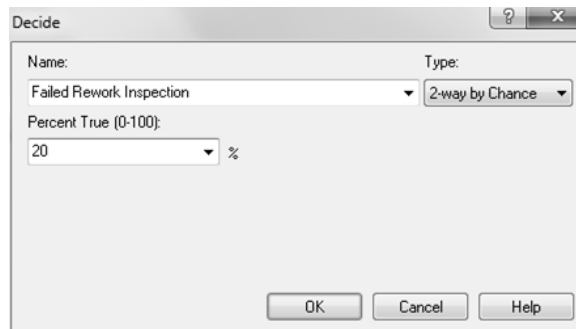
Display 4-7. Failed Sealer Inspection

Rework Process: (Display 4-8) the parts failing the inspection undergo this rework activity, set in a similar way to the Sealer Process module but changing the Name, Resource Name, and Expression (this module follows an exponential distribution with a mean of 45 minutes)



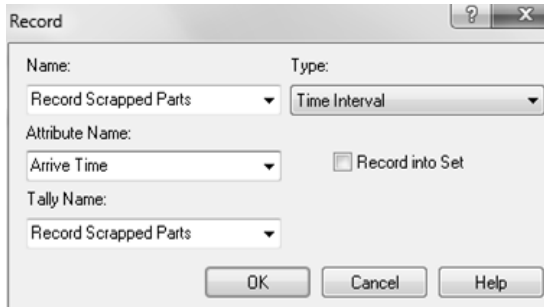
Display 4-8. Rework Process

Failed Rework Inspection: (Display 4-9) this Decide module separates the salvaged and scrapped parts after the rework operation. The parts following the True branch are the scrapped (20%) and the parts following the False branch represent the salvaged parts.

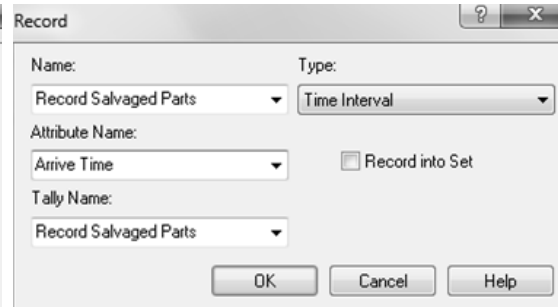


Display 4-9. Failed Rework Inspection

Record Scrapped Parts: (Display 4-10) the record module provides the ability to collect these cycle times in the form of Tallies. We choose Time Interval Type from the list; choosing this option Arena records as a Tally statistic the time each entity remains in the system (it records the time an entity arrives at the Record module and the Arrive time).



Display 4-10. Record Scrapped Parts



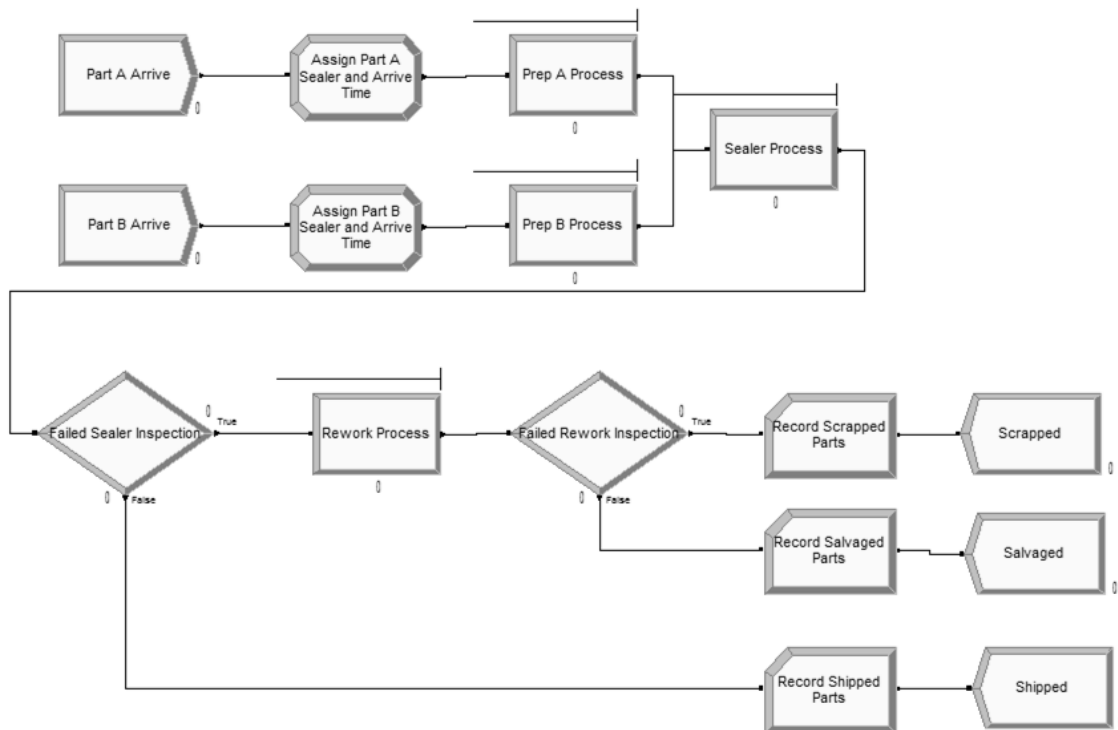
Display 4-11. Record Salvaged Parts

Record Salvaged Parts: (Display 4-11) it is completely analogous to the previous Record module

Record Shipped Parts: is the same as the other two Record modules, except for the name.

Dispose: these three modules differ only in the name (respectively Scrapped, Salvaged and Shipped).

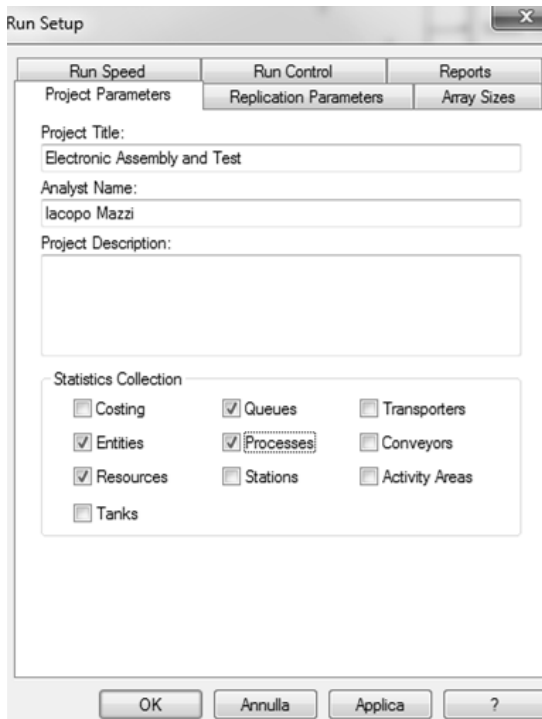
The completed model should look like Display 4-12 now.



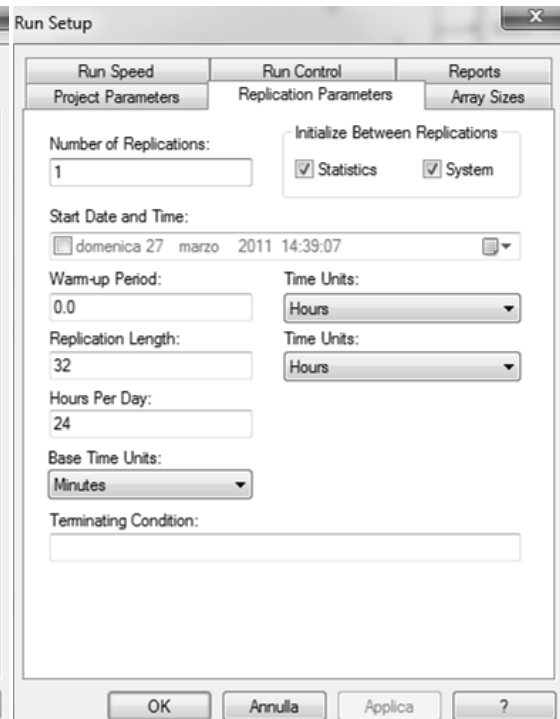
Display 4-12. Final Outlook

4.1.3. Running the Model.

If we run this model now Arena will perform the simulation forever because no duration has been set. In the Run > Setup menu is possible to establish the run parameters. In the Project Parameters tab we enter the Title, the Analyst and a Description of the project apart from the statistics of interest. In the Replication Parameters we set the number of replications (1), the replication length (32 hours) and the base time unit (minutes). The screenshots are visible in Display 4-13 and Display 4-14.



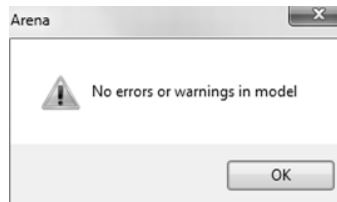
Display 4-13. Project Parameters



Display 4-14. Replication Parameters

To distinguish Part A from Part B is possible to assign a different picture to each entity. Selecting the Entity data module in the Basic Process panel is possible to choose the preferred picture for Part A (in this case the blue ball) and Part B (the red ball).

In addition, before running the model it is be important to check it for errors. It is possible to check the created model by clicking the Check button on the Run Interaction toolbar, the Run > Check Model command or the F4 key on the keyboard. In this case the response is a little window with the message "No errors or warnings in model" (Display 4-15).



Display 4-15. Positive Model Check

If the model check results in no errors it is possible to run the simulation, using one of the different possible ways to do it: Clicking the Go button (▶) on the Standard toolbar, the Run>Go command, or the F5 key. If the model has not been checked yet Arena will first check it, then initialize the model with the data and finally run it.

After the simulation starts to run it is possible to increase or decrease the speed of the model animation by pressing respectively the “>” or the “<” key. During the simulation run it is also possible to pause the simulation using the Pause button on the standard toolbar, Run > Pause or the Esc key. It is also possible to move entities through the system one step at a time by using the Step Button; after that is possible to continue the simulation run normally at any time clicking the Go button again.

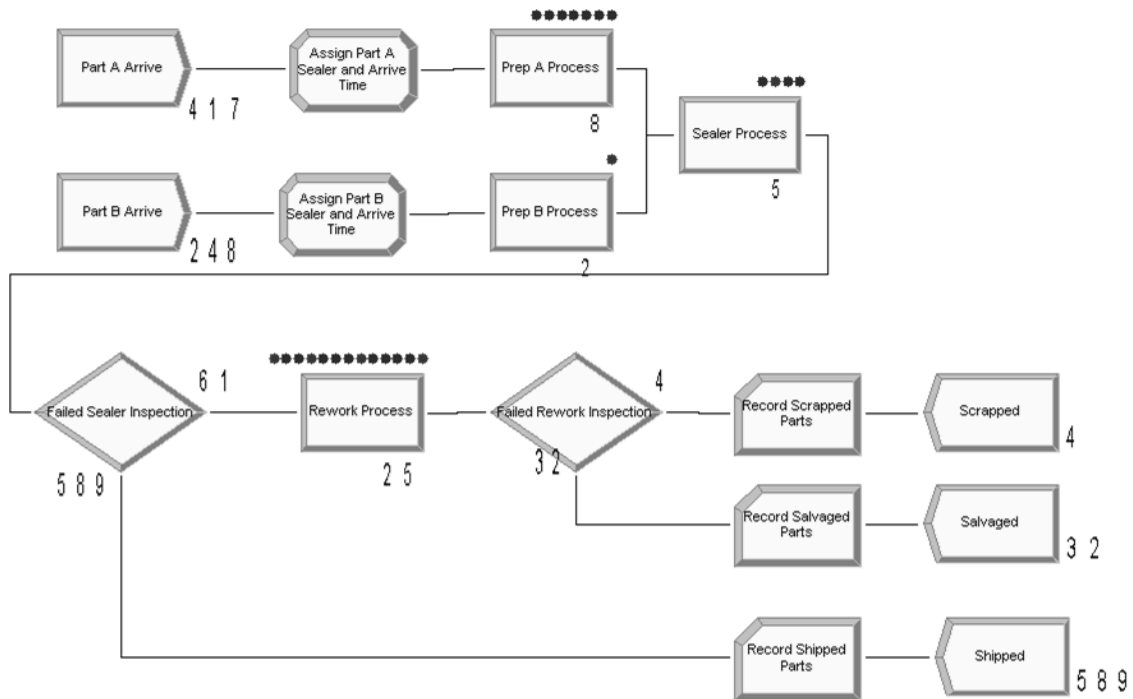


Display 4-16. Run Toolbar

The Run>Run Control option allows to configure a number of other runtime options like the Batch Run (No Animation) option, which lets the simulation run much faster. The only disadvantage is that it is necessary to terminate the run and reset the animation settings in order to get the animation back. This is the best option if the model is large and the analyst is interested only in the numerical results (Display 4-16).

4.1.4. Viewing the Results

Running the model to completion, Arena will ask if the analyst wants to see the results. Selecting Yes Arena opens the default report (Category overview Report). For this model the software shows sections on Process, Queue, Resource and User Specified. The User Specified section is present because creating the model Record Modules have been included.



Display 4-17. Animated Results

At the end of a single-replication simulation run, Arena attempts to calculate a 95% confidence-interval half width for the steady-state (long-run) expected value of each observed statistic, using a method called batch means. Arena first checks to see if sufficient data have been collected to test the critical statistical assumption (uncorrelated batches) required for the batch-means method. If not, the annotation “Insufficient” appears in the report and no confidence-interval half width is produced. If there are enough data to test for uncorrelated batches, but the test is failed, the annotation “Correlated” appears. If there was enough data to perform the test for uncorrelated batches and the test were passed, the half width of a 95% confidence interval on the long-run (steady-state) expected value of the statistic would be given (Display 4-18).

Queue

Time

Waiting Time	Average	Half Width	Minimum Value	Maximum Value
Prep A Process.Queue	14.6220	(Correlated)	0.00	46.3460
Prep B Process.Queue	26.9037	(Insufficient)	0.00	85.9510
Rework Process.Queue	456.35	(Insufficient)	0.00	810.98
Sealer Process.Queue	2.5153	(Correlated)	0.00	14.0456

Other

Number Waiting	Average	Half Width	Minimum Value	Maximum Value
Prep A Process.Queue	3.1680	(Correlated)	0.00	11.0000
Prep B Process.Queue	3.5017	(Insufficient)	0.00	14.0000
Rework Process.Queue	12.9534	(Insufficient)	0.00	26.0000
Sealer Process.Queue	0.8631	0,334935455	0.00	6.0000

Display 4-18. The Queue Section of the Category Overview Report

4.2.The Enhanced Electronic Assembly and Test System

4.2.1.System Description

In this section the previous model will be edited to introduce more details so its behavior will be more similar to a real world system.

This new system operates two eight-hour shifts a day for 10 days and on the second shift of every day two operators instead of one are assigned to the rework operation.

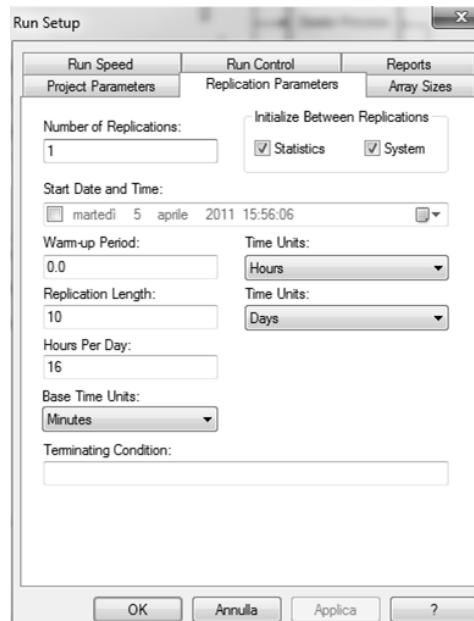
Periodically the sealer machine breaks down but no correction will be made because analysts found that the sealer operation is not a bottleneck of the system. The mean time from the end of one failure and the onset of the next one is 120 minutes and the distribution of the uptime is exponential. The repair time follows an exponential distribution too with a mean of 4 minutes.

The factory manager purchased special racks for the rework area that can hold ten assemblies each while they are in queue for the rework operation; the question is how many racks the buffer needs.

4.2.2 Schedules and States

To define the new two eight-hours shifts day open the Replication Parameters tab of the Run > Setup option and change the Hours Per Day from 24 to 16 (as shown in Display 4-19) and the Replication Length to 10 (obviously changing the Time Unit to Days).

To define the Resource Schedule open the Resource data module on the left side of the screen and change Rework Type to Based on Schedule and enter the Schedule Name, in this case Rework Schedule.

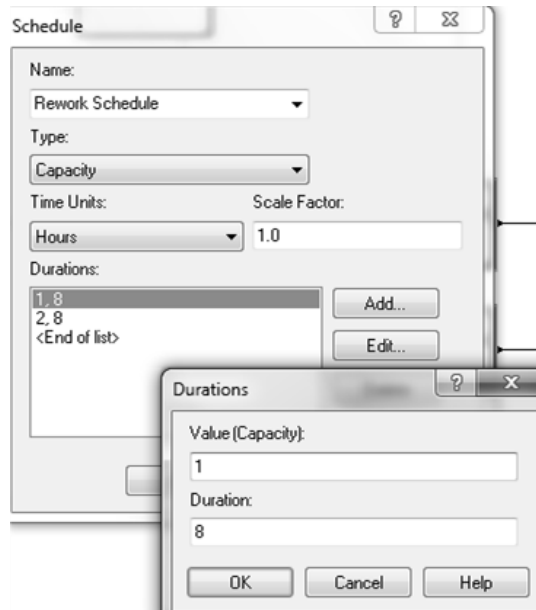


Display 4-19. New Replication Parameters

The Schedule Rule can be chosen from three options in the pull-down:

- Ignore: this option immediately decreases the resource capacity, ignoring the fact that the resource is currently allocated to an entity, but “work” on the in-service entities continues unabated. When units of the resource are released by the entity these units are placed in the Inactive state. The net effect is that the time for which the resource capacity is scheduled to be reduced may be shortened with this option.
- Wait: this option will wait until the in-process entities release their units of the resource before starting the actual capacity decrease. Thus the reduced capacity time will always be of the specified duration, the time between these reduction may increase.
- Preempt: it attempts to preempt the last unit of the resource seized by taking it away from the controlling entity. If the preempt is successful and a single unit of capacity is enough, then the capacity reduction starts immediately. If the preempt is unsuccessful or if more than one unit is needed, then the Ignore rule will be used for any remaining capacity.

We have selected the Ignore option because an operator will finish his task before leaving.



Display 4-20. Rework Schedule

To define the schedule rule right click on the Rework Schedule row in the Schedule data module and choose Edit via Dialog option. As shown in Display 4-20 add two rows: the first with capacity of 1 and duration of 8 (hours) and the second with capacity of 2 and duration of 8 (hours) too. The data entered for Day 1 will be repeated automatically for every day of the simulation.

4.2.3.Resource Failures

The Failure data module can be found in the Advance Process panel (if is not present in the Project Bar just attach it). Double click on the Failure data module to add a row.

The name will be set to Sealer Failure while the type can be set to Count or Time. A Count-based failure causes the resource to fail after the specified number of entities have used the resource while in our system the failure is a Time-based failure. Up Time and Down Time are respectively EXPO(120) and EXPO(4) (exponential distributions with mean of 120 and 4 minutes) while the Time Units are set to Minutes.

Resource - Basic Process							Failures	
	Name	Type	Capacity	Schedule Name	Schedule Rule	Busy / Hour	Failure Name	Failure Rule
1	Prep A	Fixed Capacity	1	1	Wait	0.0	1 Sealer Failure	Wait
2	Prep B	Fixed Capacity	1	1	Wait	0.0		
3	Sealer	Fixed Capacity	1	1	Wait	0.0		
4	Rework	Based on Schedule	Rework Schedule	Rework Schedule	Ignore	0.0		

Double-click here to add a new row.

Display 4-21. Sealer Resource Failure

Now we need to attach it to the Sealer resource: back in the Basic Process panel open the Resource data module, click in the Failures column and select Sealer Failure from the list as shown in Display 4-21.

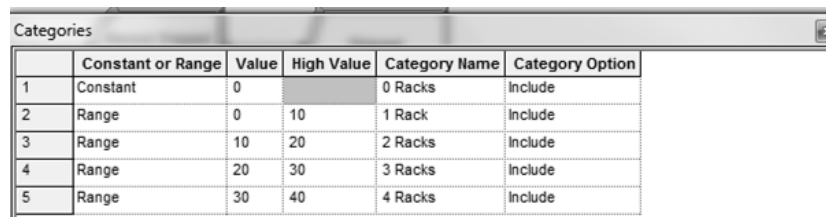
4.2.4.Frequencies

To obtain the required statistics about the number of racks needed at the Rework area we use Frequencies. In this case each rack can handle 10 items so we are interested in the amount of time the number in queue is: 0, more than 0 but no more than 10, more than 10 but no more than 20, more than 20 but no more than 30 or more than 30 but no more than 40 to establish if no rack, 1 rack, 2 racks, 3 racks or 4 racks are needed.

Frequency statistics are entered using the Statistic data module in the Advanced Process panel. Double click to add a new row: the name is Rework Queue Stats, the Type is Frequency and Frequency Type is Value. The Expression is NQ(Rework Process.Queue).

Click now on Categories and add 5 rows as shown in Display 4-22.

To show the amount of time the Sealer resource is in a failed state just add a new row in the Statistic data module. Name is Sealer States, Type is Frequency, Frequency Type is State and Resource Name is Sealer.



	Constant or Range	Value	High Value	Category Name	Category Option
1	Constant	0		0 Racks	Include
2	Range	0	10	1 Rack	Include
3	Range	10	20	2 Racks	Include
4	Range	20	30	3 Racks	Include
5	Range	30	40	4 Racks	Include

Display 4-22. Categories for Rework Queue Stats

4.2.5.Results

Table 4-2 compares some selected results from the basic model and from thin new enhanced model whereas Display 4-23 from Report > Frequencies (in the Project Bar) shows the racks required and the Sealer States previously defined.

Result	Basic Model	Enhanced Model
Average Waiting Time in Queue		
Prep A	14.62	19.20
Prep B	26.90	51.42
Sealer	2.52	7.83
Rework	456.35	116.25
Average Number Waiting in Queue		
Prep A	3.17	3.89
Prep B	3.50	6.89
Sealer	0.86	2.63
Rework	12.95	3.63
Average Time in System		
Shipped Parts	28.76	47.36
Salvaged Parts	503.85	203.83
Scrapped Parts	737.19	211.96
Instantaneous Utilization of Resource		
Prep A	0.9038	0.8869
Prep B	0.7575	0.8011
Sealer	0.8595	0.8425
Rework	0.9495	0.8641
Scheduled Utilization of Resource		
Prep A	0.9038	0.8869
Prep B	0.7575	0.8011
Sealer	0.8595	0.8425
Rework	0.9495	0.8567

Table 4-2. Selected Results

16:57:48		Frequencies			april 05, 2011	
Electronic Assembly and Test				Replications: 1		
Replication 1		Start Time: 0,00	Stop Time: 9,600,00	Time Units: Minutes		
Rework Queue Stats		Number Obs	Average Time	Standard Percent	Restricted Percent	
0 Racks		41	69.4722	29.67	29,67	
1 Rack		52	119.96	64.98	64,98	
2 Racks		12	42.8210	5.35	5,35	
Sealer States		Number Obs	Average Time	Standard Percent	Restricted Percent	
BUSY		697	11.6044	84.25	84,25	
FAILED		68	4.1861	2.97	2,97	
IDLE		640	1.9173	12.78	12,78	

Display 4-23. Frequencies Report

CHAPTER 5: A REALISTIC MODEL

5.1. System Description

The system consists of part arrivals, four manufacturing cells (Welding -Cell 1-, Hardening -Cell 2-, Drilling –Cell 3- and Pressing –Cell 4-) and a post-processing department. Cells 1 and Cell 2 have a single machine each while Cell 3 has two drills; one of the two drills at Cell 3 is a newer model that can process parts in 75% of the time required by the older drill. Cell 4 has two identical presses and 2% of pressed parts require to be pressed again.

This system produces four different part types, the part steps and process time (expressed in minutes and following a triangular distribution) are given in Table 5-1. Process times given at Cell 3 are referred to the older and slower drill.

The interarrival times between successive part arrivals are exponentially distributed with a mean of 11 minutes and the first part arrives at time 0. The distribution by type is 20% Part 1, 35% Part 2, 18% Part 3 and 27% Part 4.

After the part has been processed it undergoes a digital Quality Control process that requires 10 minutes: 85% pass this Quality Control and are sent to the Shipping Station while the remaining 15% is divided between rework station (70% of them) and recycle station (30% of them).

At the rework process work Jim, Anna, Paul and Dwayne who follow their schedules (their timetables will be shown later). Rework process requires between 9 and 15 minutes following an uniform distribution. The recycle process follows an uniform distribution too but between 20 and 25 minutes.

At the shipping station the part is put in a batch of 2 (batches of the same kind of product) and then boxed in 3 minutes.

Parts are then moved from a location to the next of its working process by a cart (there are 3 of them in the factory) that can transport one part each with a speed of 65 feet per minute and requiring 35 seconds to load or unload the part.

The distance between each part of the system is shown in Figure 5-1 where “C” means Part Creation, “1” means Cell 1, “2” means Cell 2, “3” means Cell 3, “4” means Cell 4, “QC” means Quality Control, “S” means Shipping, “RW” means Rework and “RC” means Recycle.

Part Type	Cell/Time	Cell/Time	Cell/Time	Cell/Time	Cell/Time
1	1 6, 8, 10	2 4, 8, 9	3 16, 18, 25	4 3, 7, 11	
2	1 10, 12, 16	2 3, 6, 9	3 25, 33, 38	4 12, 16, 18	2 5, 8, 10
3	2 7, 9, 12	1 8, 11, 15	3 17, 21, 25	4 7, 13, 16	
4	3 13, 16, 19	1 5, 10, 13	4 6, 11, 15	2 7, 10, 13	

Table 5-1. Part Routings and Process Times

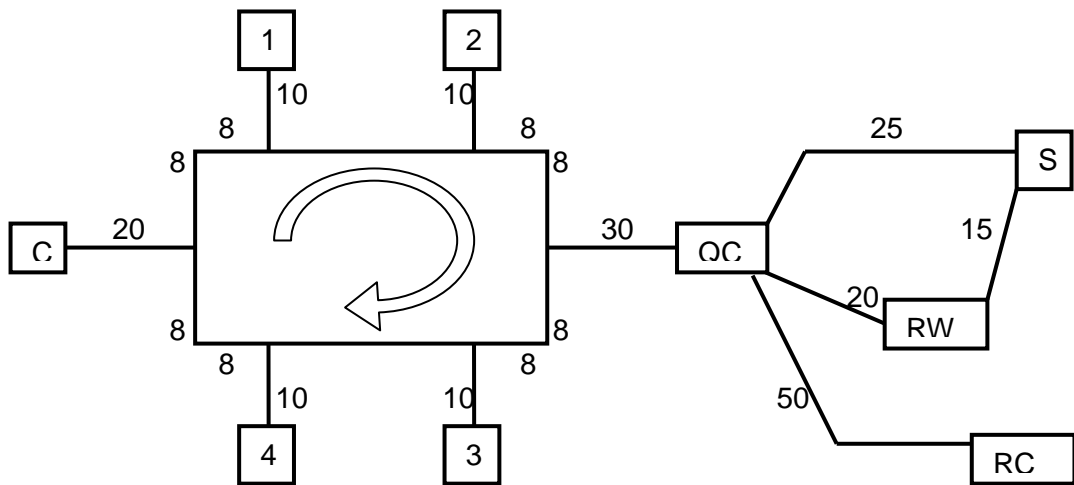


Figure 5-1. System Layout

5.2. Modeling Approach

The first, most important, peculiarity of this system is that it has to work on four different parts that follow different process plans through the system so we need a process plan with an automatic routing capability.

The second characteristic is that in Cell 3 there are two different machines that have different process times so we need to distinguish between these two machines.

The Sequence data module, on the Advanced Transfer panel, allows to define an ordered list of stations that can include assignments of variables or attributes at each station.

5.3. Building the Model

5.3.1. Data Modules

Sequence Data Module: in the Advanced Transfer panel (as usual, if it is not present in the Project Bar just attach it) double click to add a new row and enter the name of the first sequence, Part 1 Process Plan. Now enter the process steps adding a row for each station Part 1 has to pass through and Quality Control as the last row otherwise you will get a run-time error when the first entity completes its process route and Arena is not told where to send it next. For Part 1 the steps are Cell 1 (step name is Part 1 Step 1), Cell 2 (step name is Part 1 Step 2), Cell 3 (step name is Part 1 Step 3), Cell 4 (step name is Part 1 Step 4), Quality Control (step name is Part 1 Step 5). To each of these steps add an assignment: set Assignment Type to attribute, Attribute Name to Process Time and Value to $TRIA(\dots, \dots, \dots)$ as shown previously in Table 5-1. We will use an Expression to define process times for Cell 1 so Assignments have to be made only for Cell 2, Cell 3 and Cell 4. In Display 5-1 is shown the procedure for Part 1 Process Plan, Part 1 Step 2 and Process Time.

The screenshot shows the 'Sequence - Advanced Transfer' window. It contains two main tables: 'Steps' and 'Assignments'.

Steps Table:

	Name	Steps
1	Part 1 Process Plan	5 rows
2	Part 2 Process Plan	6 rows
3	Part 3 Process Plan	5 rows
4	Part 4 Process Plan	5 rows

Steps Table (Detailed):

	Step Name	Next Step	Assignments
1	Part 1 Step 1		0 rows
2	Part 1 Step 2		1 rows
3	Part 1 Step 3		1 rows
4	Part 1 Step 4		1 rows
5	Part 1 Step 5		

Assignments Table:

	Assignment Type	Attribute Name	Value
1	Attribute	Process Time	TRIA(4, 8, 9)

Display 5-1. Sequence Data Module

Expression - Advanced Process							Expression Values	
	Name	Rows	Columns	Data Type	File Name	Expression Values		
1	Cell 1 Times	4		Native		4 rows	1	TRIA(6, 8, 10)
2	Reworker Times	3		Native		3 rows	2	TRIA(10, 12, 16)
							3	TRIA(8, 11, 15)
							4	TRIA(5, 10, 13)

Double-click here to add a new row.

Display 5-2. Expression Data Module

Expression Data Module: in the Advanced Process panel we define the expression for Cell 1 process time, called Cell 1 Times. These process times could have been put previously in the Sequence data module but this is just a different way to operate. As shown in Display 5-2 Cell 1 Times Expression contains four rows, the four different triangular distributions for each part at Cell 1.

Variable Data Module: in this Data Module we define Factor Variable (to describe that the newer drill processes parts in 75% of the time required by the older one). As shown in Display 5-3 the new drill will be referenced as 1 and its factor will be 0.75 (to describe that it works in 75% of the time) and the old drill will be referenced as 2 with a factor of 1.0.

Variable - Basic Process								Initial Values	
	Name	Rows	Columns	Data Type	Clear Option	File Name	Initial Values		
1	Factor	2		Real	System		2 rows	1	0.75
								2	1.0

Double-click here to add a new row.

Display 5-3. Drill Factors

Set Data Module: in this data module we will form sets for Cell 3 drills, Cell 4 presses, part pictures, entity types and rework operators. The first Resource set contains Cell 3 New and Cell 3 Old as members; the Cell 4 Machines has Press A and Press B as members; the Entity Picture set named Part Pictures contains Picture.Gear 1, Picture.Gear 2, Picture.Gear 3 and Picture.Gear 4 as members; the Entity Type set is named Entity Types and has four members: Part 1, Part 2, Part 3 and Part 4 and the Rework Operators set has Jim Anna Paul and Dwayne as members as shown in Display 5-4.

Set - Basic Process				Members		Members		Members		Members	
	Name	Type	Members		Resource Name		Picture Name		Entity T		Resource Name
1	Cell 3 Machines	Resource	2 rows	1	Cell 3 New	1	Picture.Gear 1	1	Part 1	1	Jim
2	Cell 4 Machines	Resource	2 rows	2	Cell 3 Old	2	Picture.Gear 2	2	Part 2	2	Anna
3	Entity Types	Entity Type	4 rows			3	Picture.Gear 3	3	Part 3	3	Paul
4	Part Pictures	Entity Picture	4 rows			4	Picture.Gear 4	4	Part 4	4	Dwayne
5	Rework Operators	Resource	4 rows								

Double-click here to add a new row.

Display 5-4. Set Data Module

Advanced Set - Advanced Process			
	Name	Set Type	Members
1	Part Sequences	Other	4 rows

Double-click here to add a new row.

Members	
	Other
1	Part 1 Process Plan
2	Part 2 Process Plan
3	Part 3 Process Plan
4	Part 4 Process Plan

Display 5-5. Advanced Set Data Module

Advanced Set Data Module: this Data Module, visible in the Advanced Process panel of the Project Bar lists three different types: Queue, Storage and Other. The Other option allows to form sets of any similar Arena objects. Using this option our new set, called Part Sequences, will contain three members (as shown in Display 5-5): Part 1 Process Plan, Part 2 Process Plan, Part 3 Process Plan and Part 4 Process Plan.

Schedule - Basic Process					
	Name	Type	Time Units	Scale Factor	Durations
1	Jim Timetable	Capacity	Hours	1.0	4 rows
2	Anna Timetable	Capacity	Hours	1.0	5 rows
3	Paul Timetable	Capacity	Hours	1.0	5 rows
4	Dwayne Timetable	Capacity	Hours	1.0	3 rows

Display 5-6. Schedule Data Module

Schedule Data Module: Jim, Anna, Paul and Dwayne work 8 hours a day each and in Table 5-2 are shown their daily timetables so four different timetables have been added in this module (Display 5-6). Remember to add Capacity 0 after working hours to complete the 24-hour day.

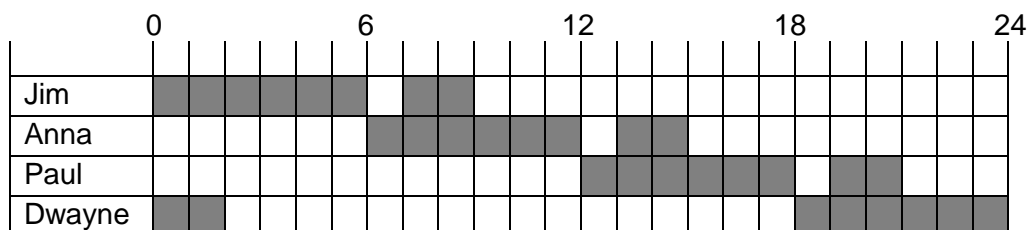


Table 5-2.

Transporter Data Module: in the Transporter Data Module we have to insert that three carts are available. Our carts are Free-Path which means that carts can move freely through the system without delays due to congestion. As shown in Display 5-7 their speed is 65 units per minute (in our example distances are expressed all in feet and speeds in feet per minute) and the initial position is at the Order Release Station.

Transporter - Advanced Transfer							
	Name	Number of Units	Type	Distance Set	Velocity	Units	Initial Position Status
1 ▶	Cart	3	Free Path	Cart.Distance	65	Per Minute	1 rows
Double-click here to add a new row.							

Initial Position Status		
	Initial Position	Initial Status
1	Station	Active
Double-click here to add a new row.		

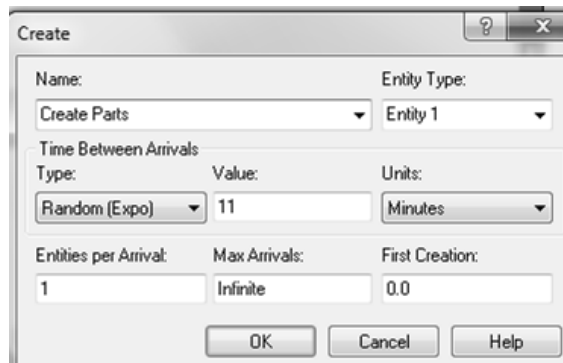
Display 5-7. Transporter Data Module.

Distance Data Module: in the Cart.Distance entry we have to patiently add all the distances from each station to all other stations of the model remembering that in the Cells part of the system carts move only in clockwise direction.

Run Setup: in the Run > Setup dialog box we change the Replication Length to 10 days and set the Base Time Units as Minutes

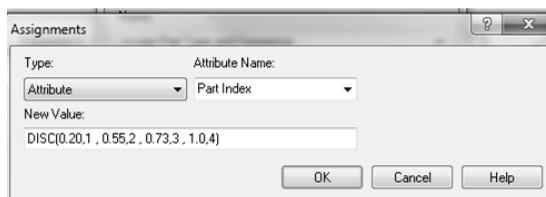
5.3.2.Logic Modules

Create Parts: this first module, named Create Parts, uses a Random distribution with a mean of 11 minutes to generate the arriving parts (as shown in Display 5-8).

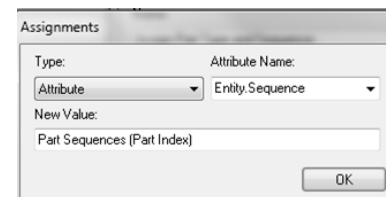


Display 5-8. Create Parts Module

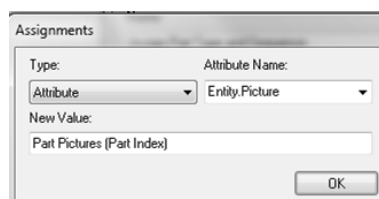
Assign Part Type and Sequence: in this module we associate a sequence to arriving entities and define an index, Part Index, for our sets that will allow us to associate the proper sequence, entity type and picture to each arrival. To determine which part type has arrived we use a Part Index Attribute following a discrete distribution with given cumulative probabilities. Entity.Sequence Attribute with Part Sequences (Part Index) value associates the created entity to the proper part sequence (all of these attributes are shown in Display 5-9, Display 5-10, Display 5-11 and Display 5-12). To associate the proper entity picture and type for the newly arrived entities we use the Entity.Picture and Entity.Type attributes.



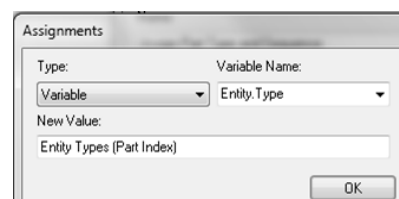
Display 5-9. Part Index



Display 5-10. Entity.Sequence

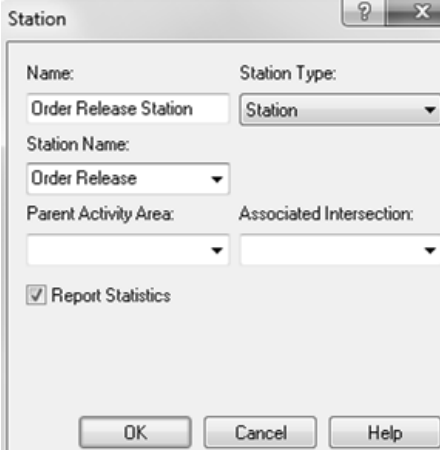


Display 5-11. Entity.Picture



Display 5-12. Entity.Type

Order Release Station: this module from the Advanced Transfer panel will be defined when we send the entity through a station module, which will define the station and tell the software that the current entity is at that location (Display 5-13).

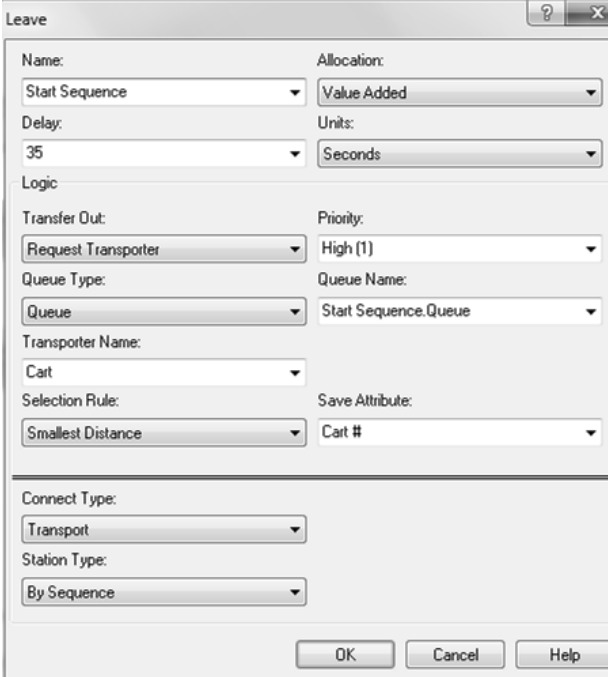


The screenshot shows a dialog box titled "Station". It contains the following fields and controls:

- Name: Order Release Station
- Station Type: Station
- Station Name: Order Release
- Parent Activity Area: (empty)
- Associated Intersection: (empty)
- Report Statistics
- Buttons: OK, Cancel, Help

Display 5-13. Order Release Station

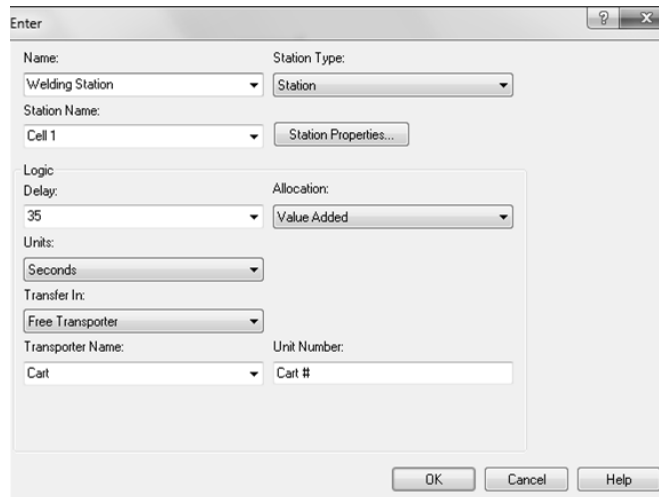
Start Sequence: (Display 5-14) this Leave module transfers an entity to the first station defined in the Process Plan. The Delay of 35 seconds means that the cart needs 35 seconds to load each part before being ready to move it. The other entries show that this module requires a cart to transfer parts and it requests the nearest idle one.



The screenshot shows a dialog box titled "Leave". It contains the following fields and controls:

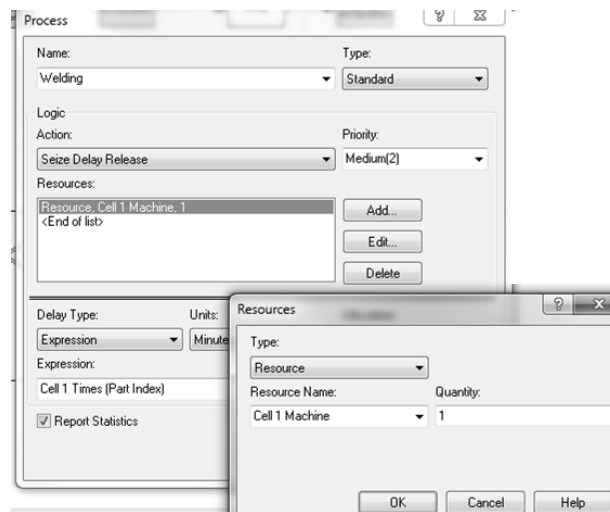
- Name: Start Sequence
- Allocation: Value Added
- Delay: 35
- Units: Seconds
- Logic section:
 - Transfer Out: Request Transporter
 - Priority: High (1)
 - Queue Type: Queue
 - Queue Name: Start Sequence.Queue
 - Transporter Name: Cart
 - Selection Rule: Smallest Distance
 - Save Attribute: Cart #
- Connect Type: Transport
- Station Type: By Sequence
- Buttons: OK, Cancel, Help

Display 5-14. Start Sequence



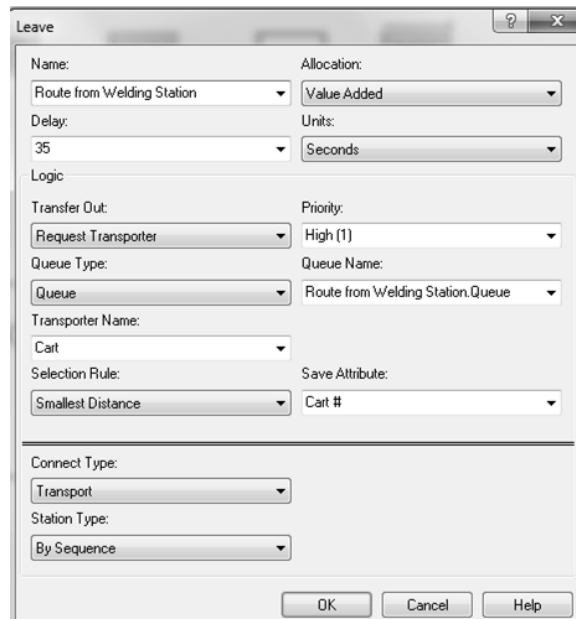
Display 5-15. Welding Station Enter module

Cell 1 Modules: the welding system is composed of an Enter module, a Process module and a Leave module. As Shown in Display 5-15 the Welding Station has been called Cell 1 and it provides the location to which a part can be sent; as described before carts need 35 seconds to unload the part and then they are free to collect other



Display 5-16. Welding Process

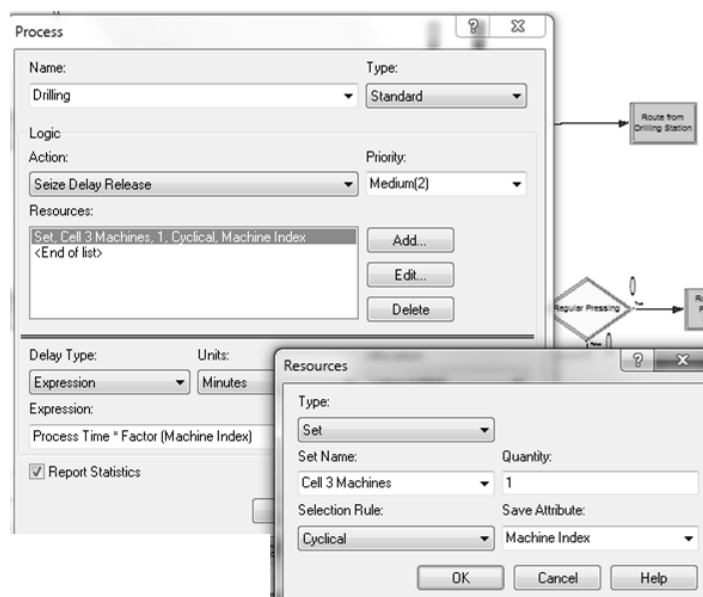
parts from other locations. Welding Process module's delay time has been previously defined in the expression so, as shown in Display 5-16, we have entered the Cell 1 Times expression using the Part Index attribute to reference the appropriate part-processing time. After the process the entity is sent to the Leave module (Display 5-17) where it is routed to the next step.



Display 5-17. Route from Welding Station

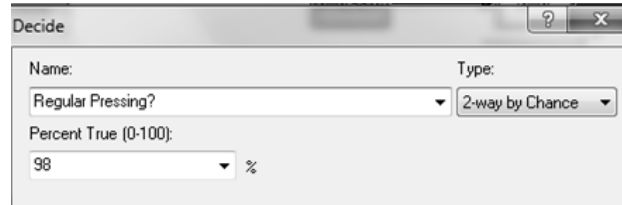
Cell 2 Modules: the three modules (Enter, Process and Leave) are very similar to Cell 1 modules with the only exception that in Hardening Process time expression we put Process Time, recalling the times previously put in the Sequence data module.

Cell 3 Modules: the Enter and the Leave modules are exactly the same of Welding and Hardening Cells but the Drilling Process is different because we have two different drills with a different process time. As shown in Display 5-18 we put Cell 3 Machines set in the resource section; the default selection rule (Cyclical) has been selected. The cyclical selection rule causes the entity to select the two resources alternately but if only one resource is available it will be selected anyway.



Display 5-18. Drilling Process

Cell 4 Modules: the pressing process has a 2% failure rate so after the Pressing Process module we put a Decide Module (Display 5-19) with a 98 Percent True. The True branch goes to the Leave module while the False branch goes to the Process module again.



Display 5-19. Decide Module

Welding Failures: the welding robot has a Down Time following an exponential distribution with a mean of 60 minutes after an Up Time of 1500 minutes following an EXPO distribution too so in the Failure data module (Advanced Process) we insert a Welding Failure row and in the Resource data module we add a failure row for Cell 1 Machine (Display 5-20).

Resource - Basic Process			Failures	Report Statistics
	Name	Type		
1	Cell 3 New	Fixed Capacity	0 rows	
2	Cell 3 Old	Fixed Capacity	0 rows	
3	Cell 1 Machine	Fixed Capacity	1 rows	
4	Cell 2 Machine	Fixed Capacity	0 rows	
5	Cell 4 Machine	Fixed Capacity	0 rows	
6	Recycle Machine	Fixed Capacity	0 rows	
7	Jim	Based on Schedule	0 rows	

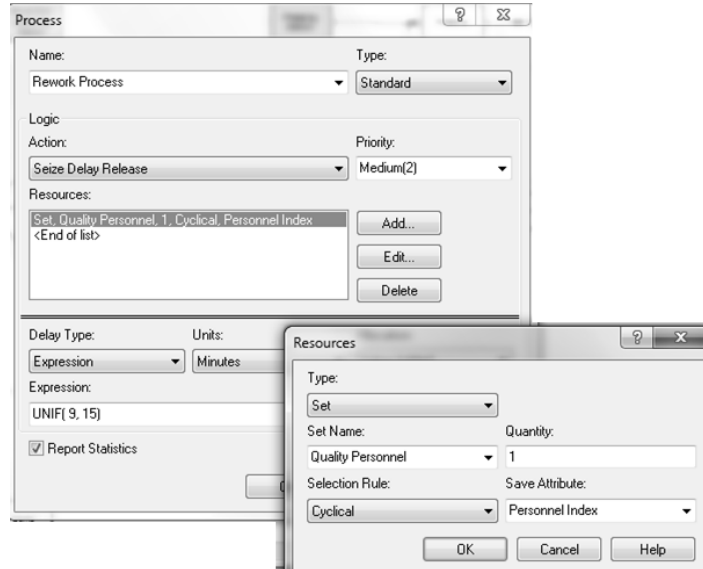
Failures	
	Failure Rule
1	Welding Failure

Display 5-20. Welding Failure

Quality Control Modules: after the Quality Control Enter module the four different parts undergo a Quality Control process with a constant process time of 10 minutes (the automatic quality control needs the same time for each part) and then is sent to a Decide module (“Control Passed?”) with an 85% success rate. If the part passes successfully the control then is sent to the Shipping station otherwise with another Decide module we see if the part is repairable or not. The second Decide module (“Repairable?”) has a 70 Percent True and the parts following the True branch are sent to the Rework Station while the parts following the False branch are sent to the Recycle Station both with a Leave module (now specifying the destination).

Rework Modules: after the usual Enter module we find the Rework Process module. In the Set data module we have put the Rework Operators set and in the Schedule data module their timetables. In the Resource data module we match each employee with his schedule choosing Based on Schedule capacity and matching each schedule

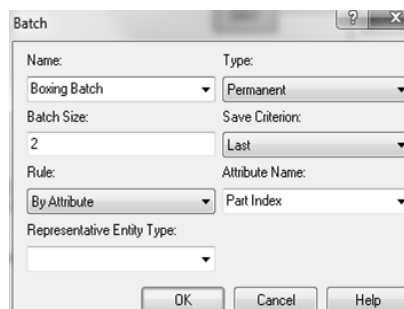
name. As shown in Display 5-21 in the Rework Process module Quality Personnel set has been added as a resource with a cyclical selection route. As described before the delay type is UNIF(9, 15) Expression. After this process reworked parts are sent to the Shipping Station.



Display 5-21. Rework Process

Recycle Modules: the Recycle Process requires an Uniform delay time with a minimum of 20 and a maximum of 25 minutes. Obviously before this Process module an Enter module has been put to receive parts from the Quality Control division. Finally a Dispose module has been added.

Shipping Modules: in this system parts are shipped in batches of 2 of the same part type so after the familiar Enter module we introduce a Batch module (Display 5-22). Entities arrive at this Batch module and are places in a queue; when two entities of the same type are accumulated, a single permanent entity leaves the module. After this module the batched entities need a 3 minutes Boxing Process before being shipped with a Dispose module.



Display 5-22. Batch Module

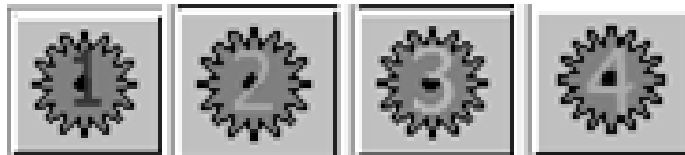
5.4. System Animation

Distances

All parts are transferred from a cell to the next of its schedule in a clockwise sense. First of all click the Station button on the Animate Transfer tool bar and locate all the 9 stations of the system. Then clicking on the Distance button (always on the Animate Transfer tool bar) draw all the routes from each station to the other eight remembering to follow the correct clockwise direction.

After having placed all the necessary distances draw the borderlines of the system creating a factory layout large enough to let the entity pictures move inside giving a realistic sensation.

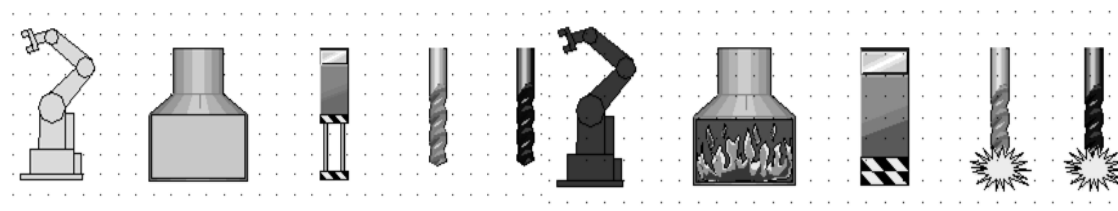
Entity Pictures: to create customized entity pictures open Edit > Entity Pictures, choose the starting picture, add it to the right column (selecting it on the left column and pressing “>>”) and double click on it. After having edited it just close the window and copy it back to the left column assigning the corresponding entity on Value. For this model Part 1 picture will be a gear with a “1” inside, Part 2 picture will be a gear with a “2” inside, Part 3 picture will be a gear with a “3” inside of it and Part 4 picture will be a gear with a “4” inside (as shown in Display 5-23).



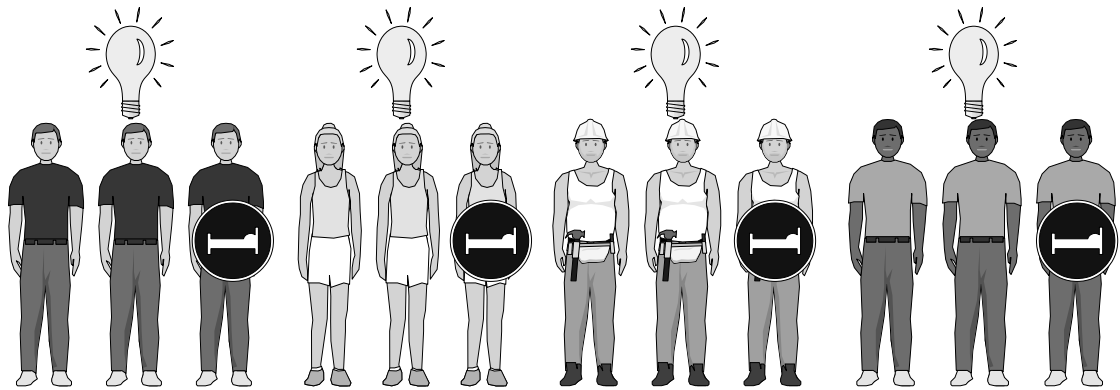
Display 5-23. Entity Pictures

Resource Pictures

From the Tools > Arena Symbol Factory choose and edit the pictures for all the 11 resources of the system and locate them near the previously located animation stations. Pictures chosen for the 6 working resources (Drill Process has two different drills whereas the two presses have the same picture because they are identical) are shown in Display 5-24 in both Idle and Busy states.

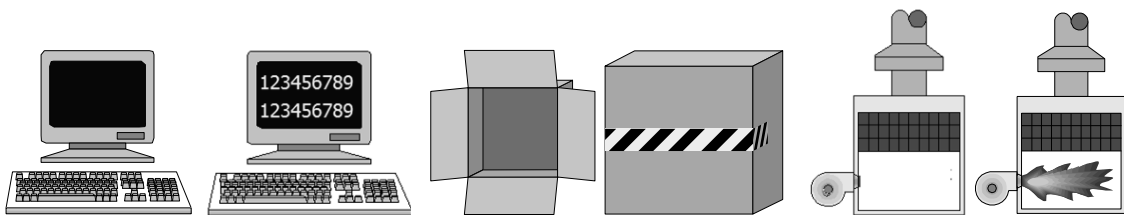


Display 5-24. Cell 1, Cell 2, Cell 3 and Cell 4 Resources Pictures



Display 5-25. Jim, Anna, Paul and Dwayne

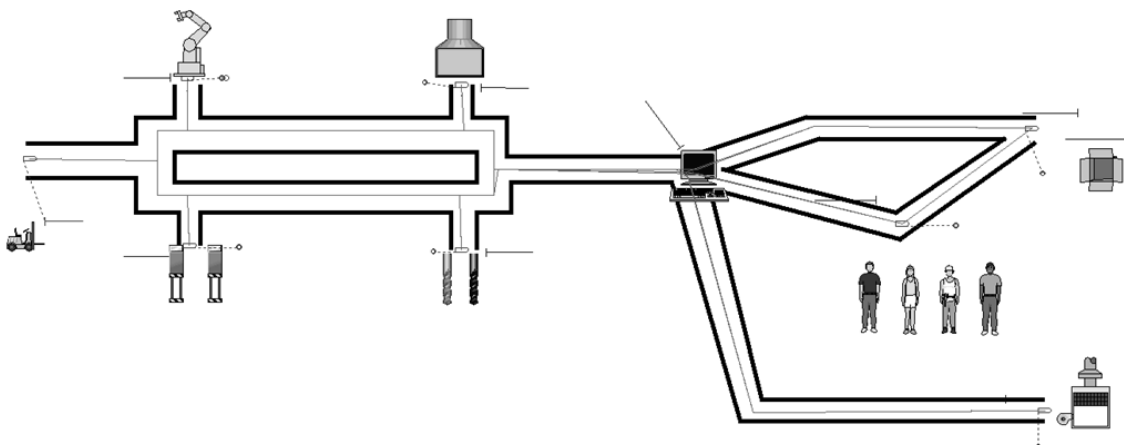
Next add pictures for the four Rework employees, the Quality Control computer, the Boxing Process and the Recycle machine. Remember that each Rework employee can be Idle, Busy or Inactive. In Display 5-25 are shown the pictures chosen for the four employees in their 3 status. In the Display 5-26 are shown pictures for the Quality Control, Boxing and Recycling resource.



Display 5-26. Quality Control, Boxing and Recycling Pictures

Now locate the Queues next to each Station and then click on the Parking button on the Animate Transfer toolbar to locate the parking for the incoming carts; to create cart pictures click on the Transporter Button, always on the Animate Transfer toolbar.

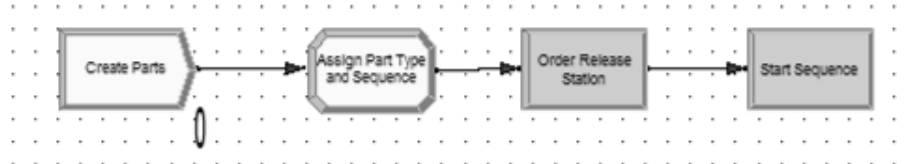
Display 5-27 shows the completed animation before running the model.



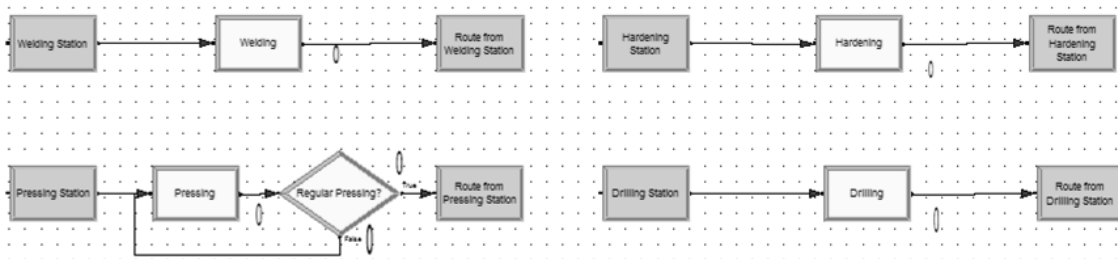
Display 5-27. Completed Animation

5.5. System Overview

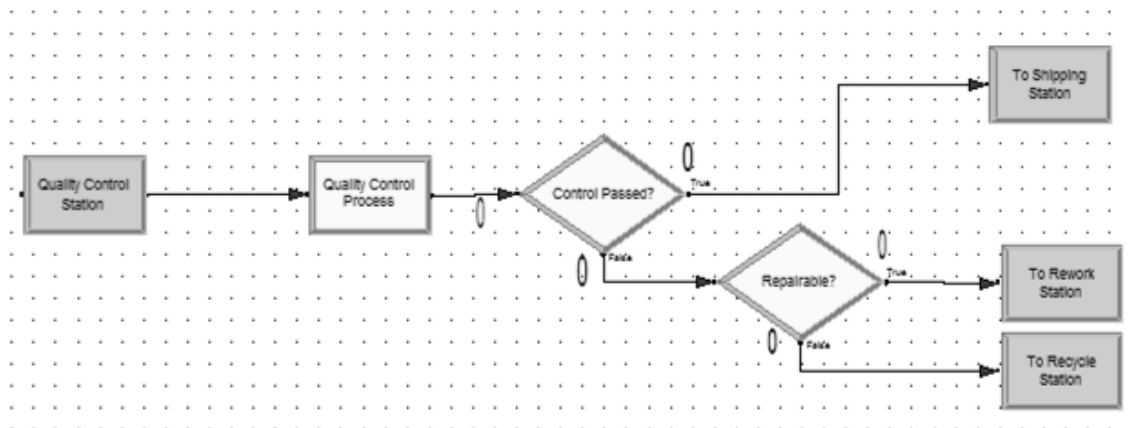
Here is shown a display for each part of the system.



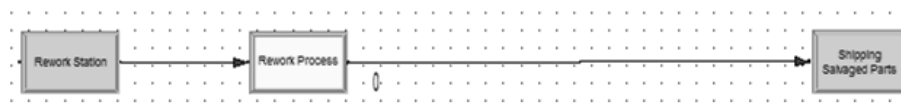
Display 5-28. Parts Creation



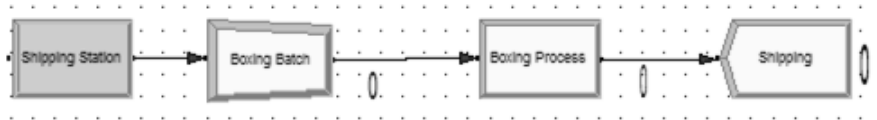
Display 5-29. Welding, Hardening, Drilling and Pressing



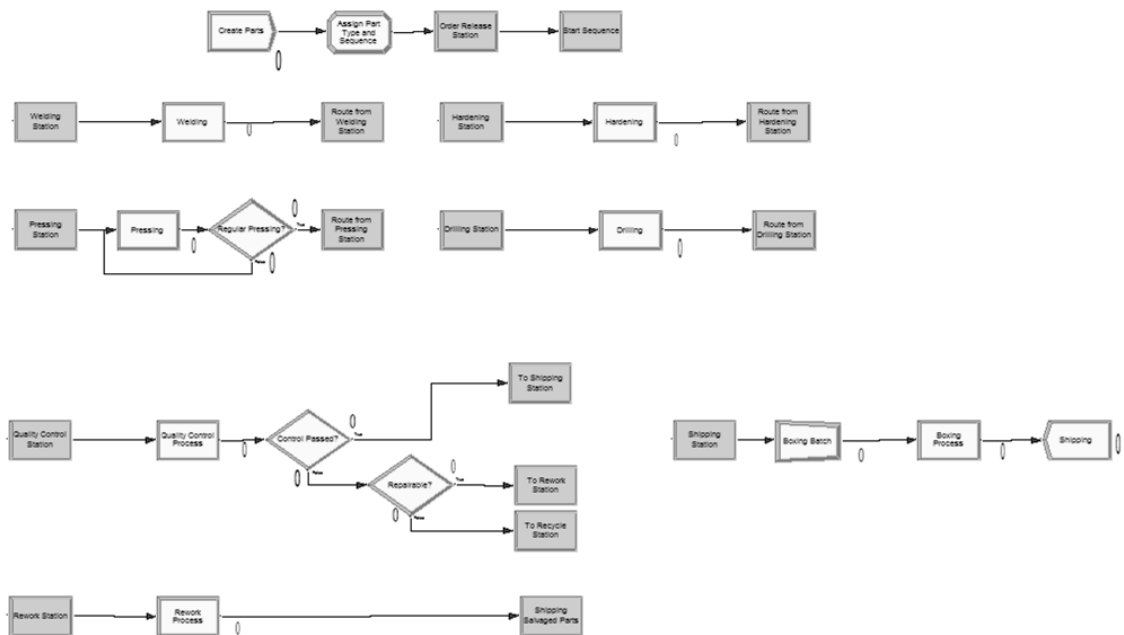
Display 5-30. Quality Control



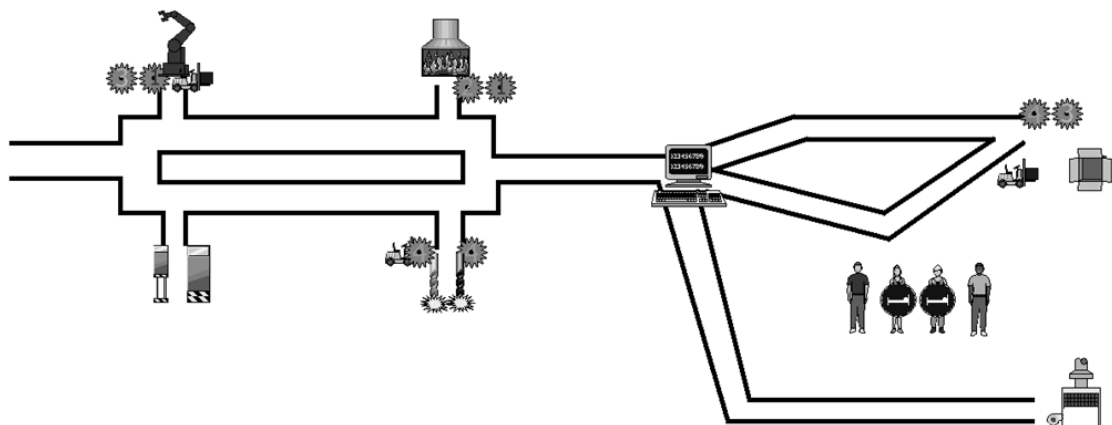
Display 5-31. Rework



Display 5-32. Boxing and Shipping



Display 5-33. Complete System



Display 5-34. Running Animation

CONCLUSION

Arena's potentials are extremely high, thank to an user friendly interface and to the flexible model creation system: from logistic to transport networks, from medical services to product distribution, from financial institutes to warehouse management.

Graphic elaborations help to represent the most complex schemes and simulation results are provided in the clearest way possible helping even the most inexperienced user to understand them and to detect critical areas like bottlenecks or insufficient usage of resources

During this work fundamental elements have been introduced to make up a model in Arena, and then these elements were analyzed in detail during the creation of realistic models, from basic ones to a more complex and detailed simulation.

The most important advantages in using Arena software are:

- Affordable price
- Software flexibility
- Easiness to learn Arena language, model creation and results comprehension
- Schematic representation of simulation results
- Widespread of the software in industrial field
- Integration with different programming languages.

Summarizing Arena gives maximum flexibility and allows to create a complete and realistic model of any industrial field, at an affordable price and with extremely complete and incisive results.

REFERENCIES

- W. David Kelton, Randall P. Sadowski, Nancy B. Swets, 2010, Simulation With Arena – Fifth Edition, McGraw-Hill International Edition
- Giorgio Romanin Jacur, Queueing Theory, issue.
- Object Database Management Systems (www.odbms.org)
- ACT Solutions (www.actsolutions.it)
- Arena Simulation Software (www.arenasimulation.com)
- Wikipedia (<http://it.wikipedia.org>)

ACKNOWLEDGEMENTS

I would like to thank all the people who supported me during these pleasant, but sometimes difficult, years.

First of all I would like to express my gratitude to my supervisor, Giorgio Romanin Jacur, whose enthusiasm and love for his subject has infected me.

It is not possible to forget the precious help given by Michela and Mauro during these months.

A special thank to my friends, and especially Alessandro, Chiara, Giulia, Marco, Mattia, Roberto and Valentina: their loyal friendship and moral support helped me to overcome the obstacles on my path.

This thesis is dedicated to my parents, Mario e Nadia, without whom none of this would have been even possible, and to my grandparents for their wholehearted love.

If you can't explain it simply,
you don't understand it well enough.
(Albert Einstein)