

# UNIVERSITY OF CINCINNATI

---

DATE: 04/28/2005

**I, Ameya Shrivastava**

hereby submit this work as part of the requirements for the degree of:

**Master of Sciences**

in:

**Industrial Engineering**

It is entitled:

**Simulation Modeling of Warehouse Environment  
utilizing AGVS Material Handling Systems**

This work and its defense approved by:

Chair: **Dr. Ernest L. Hall**  
**Dr. Richard L. Shell**  
**Dr. Ronald L. Huston**

**Simulation Modeling of Warehouse Environment utilizing AGVS Material Handling Systems.**

A Project submitted to the Division of  
Graduate Studies and Advanced Research  
Of the University of Cincinnati

In partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

In the Department of Mechanical, Industrial and Nuclear Engineering  
Of the College of Engineering  
2005

by

**Ameya Shrivastava**

B.E (Mechanical Engineering),

Y.C.C.E, Nagpur University, India, 2001.

**Thesis Advisor and Committee Chair: Dr. Ernest L. Hall**

## **Abstract**

Simulation is a very powerful and flexible tool in the design and analysis of many different types of systems. Discrete-event simulation is commonly used tool in warehousing and manufacturing industry. Although it is a very powerful and flexible means for designing and analyzing many different types of systems, without a systematic approach, it can be inefficient, expensive, and even misleading. However, a close examination of the steps one has to take through real-life applications shows that a common methodology can be used to successfully utilize simulation in all application areas. This project presents a simulation model for the warehouse operations. The warehouse operations comprising an AGV system is simulated for a random environment. The randomness is created in the system due to varying demand and the service time. ARENA 7.0 is used to develop a discrete simulation model. The significance of the study is to interpret this simulation technique that will enable the management to make better decisions.



## **Acknowledgements**

I would like to thank Dr. Hall for his insight and patience throughout the process of creating the models. He has been really inspiring figure all this time and without his guidance and support this Masters Project would not have been possible.

I would like to thank Dr. Richard L. Shell and Dr. Ronald L. Huston for being on the defense committee. I would also like to thank them for expressing their valuable comments and feedbacks.

The Robotics Team at UC has been really supportive throughout the graduate studies and I would like to thank them for keeping me motivated. Finally I would like to thank my family and friends for all of their support in my endeavors.

## Table of Contents

Abstract	
Acknowledgements	
<b>Chapter 1</b>	
1.1 Introduction	7
1.2 Overview of Available Software	10
1.3 Reason for Selecting Arena 7.0 Environment	12
<b>Chapter 2</b>	
2.1 AGV Introduction	14
2.2 Major categories of AGV	15
<b>Chapter 3</b>	
3.1 Simulation of AGV systems	16
3.2 The Warehouse System	18
<b>Chapter 4</b>	
4.1 Simulation of the System	19
4.1.1 Arrival Rate	19
4.1.2 Input modeling	20
<b>Chapter 5</b>	
5.1 Simulation Concepts	23
5.1.1 Basic Processes	23
5.1.2 Entities and Attributes	25
5.1.3 Queues	26
5.1.4 Resources	26
5.1.5 Statistics	28
<b>Chapter 6</b>	
6.1 Arena Model	29
6.2 Model components	30
6.2.1 Order Creation	30
6.2.2 Order Processing Time	32
6.2.3 AGV	33
6.2.4 Packaging	34
6.2.5 Retrieved order	35
6.3 Setting the Run Conditions	36
6.4 Running the model	37
<b>Chapter 7</b>	
7.1 Siman Code	37
<b>Chapter 8</b>	
Arena Output Results	41
8.1 Entities	41
8.1.1 Value Added Time	41
8.1.2 Non Value Added Time	42
8.1.3 Wait Time	42
8.1.4 Transfer time	42
8.1.5 Other Time	42
8.1.6 Total Time	42

8.1.7 Number in.....	42
8.1.8 Number out.....	43
8.1.9 WIP.....	43
8.1.10 Waiting time.....	43
8.1.11 Number Waiting.....	43
8.2 Resources.....	44
8.2.1 Instantaneous utilization.....	44
8.2.2 Number busy.....	44
8.2.3 Number Scheduled.....	45
8.2.4 Scheduled utilization.....	45
8.2.5 Total number seized.....	46
<b>Chapter 9</b>	
9.1 Summary and Conclusions.....	47
9.2 References.....	48

### Table of Figures

<b>Figure 1:</b> AGV controlled by floor-imbedded guidance wire.....	15
<b>Figure 2:</b> Conventional and Tandem AGV System.....	18
<b>Figure 3:</b> Block diagram of Warehouse system.....	18
<b>Figure 4:</b> Inter Arrival Data.....	20
<b>Figure 5:</b> Inter-arrival Scatter plot.....	21
<b>Figure 6:</b> Arena Input Analyzer plot .....	22
<b>Figure 7:</b> Project Bar with basic processes.....	24
<b>Figure 8:</b> Spreadsheet view of the model.....	24
<b>Figure 9:</b> Arena Model for the warehouse system.....	29
<b>Figure 10:</b> ‘Bronze Valve Orders Arrival’-Create.....	30
<b>Figure 11:</b> ‘Steel Valve Orders Arrival’-Create.....	30
<b>Figure 12:</b> ‘Order Processing Time’ – Process.....	32
<b>Figure 13:</b> ‘AGV’ - Process.....	33
<b>Figure 14:</b> Resources and Set.....	34
<b>Figure 15:</b> Packaging’ – Process.....	35
<b>Figure 16:</b> ‘Retrieved order ready for shipment’-Dispose.....	35
<b>Figure 17:</b> Run Setup Dialog .....	36
<b>Figure 18:</b> Play Button on the Standard Tool Bar.....	37

## **Chapter 1**

### **1.1 Introduction**

Computer simulation is a technique that allows building of and experimenting with a model of a real system on a computer. Although there are several different types of simulation, the focus of this project is on discrete-event simulation as applied to warehouse systems.

The discrete-event simulation technique is used for systems where the state changes occur when events happen at discrete points in time. For example - release of an order to shop floor, breakdown of a machine, completion of a machine cycle are events that change the states of a manufacturing system. This kind of simulation is more appropriate for systems with multiple entity types and event types. Job shops, an assembly line, a paint shop in a vehicle manufacturing plant are examples of such systems. The types of entities that can be seen in a job shop are orders, machines, and workers. Clearly, some of the entities would be considered as permanent, e.g. a machine, and some others would be temporary, e.g. a part, with respect to their lifetime in the model. Analysis of the throughput capability of the system, determination of the bottleneck operations, evaluation of different material handling systems are typical applications of discrete-event simulation models.

The overall process to be used in a simulation study starts with a problem definition. Once a problem is defined and simulation is selected as an analysis tool for solving it, data about the real system must be gathered. After collecting sufficient information, a model of the real system is built. Analyses with a simulation model generally involve extensive experimentation using various what-if scenarios. Based on such

experimentation, conclusions are drawn and a report is written to document the findings of the study.

System modeling is an important component of engineering. A model represents a system and the relationships that influence that system. This representation can take the form of physical and analogue models, such as globes or clay models, or schematic and mathematical models, such as organizational charts and equations. In general, models are used to save time and money, for training purposes, to determine how to optimize a system, to predict performance, to enhance understanding of system behavior and to examine worst-case scenarios. If experimenting with a system is not possible, then a model can be used to analyze the system. For example, no company would build two factories and analyze which one works better; rather they would design a model that represents the two different layouts, evaluate which system is better and then build one factory. Modeling also offers benefits for analyzing worst-case scenarios; many systems have disastrous consequences if a system enters a worst-case scenario. Creating a model allows users to learn about the worst-case scenario and explore alternatives and consequences without undue risk. There are many systems that can be modeled, ranging from chemical processing to economic behavior. This thesis will focus on discrete part manufacturing systems. In these systems each station processes a job, with the final result being some value-added product. There are three types of models that can be used for discrete event part manufacturing: physical, analytical and simulation. A physical model is one where the entire system is scaled down. Physical models are useful for educating people about the system, what it looks like, etc., but do not elucidate system behavior. Analytical models and simulation models therefore offer the best means of exploring and

understanding system behavior, but there exist tradeoffs between accuracy and effort. Simulation models have three key characteristics: the duration of the system analysis, the degree of randomness and the continuity of state variables. The duration of the system analysis refers to whether the system is studied at a point in time (static simulation), or if the system is studied for an extended period of time (dynamic simulation). The randomness takes into account the behavior of the input variables.

Deterministic simulations contain no randomness in the input variables, while the inputs to a stochastic simulation are probability distributions, which have an inherent variability. The continuity of the state variables deals with the possible values of the states, either defining the state of the system with a discrete set of values or with continuous variables.

A discrete-event system is one where the states of the resources and stations can be clearly defined (e.g., busy, idle, down) by discrete variables and only events can cause a change in the state of the system. An example of a discrete-event system would be a manufacturing process, where the state of the system is the state of each machine and the arrival or completion of each part will change the state of the system. In a continuous system, the state changes with respect to time and continuous variables represent the state of the system. An example would be a chemical plant, where the pressure or temperature of some component would change continuously with respect to time.

This research will focus on discrete-event simulation using the commercially available simulation program Arena.

## 1.2 Overview of Available Software

Simulation can be carried out using anyone of the various software available. With the vast selection of simulation software available in the market, it is imperative that the most competitive, accurate and cost effective one be chosen. The popular simulation software available are:

**20-sim:** 20-sim (Controllab products) is relatively new software used for blocks modeling system for continuous systems. 20-sim is a modeling and simulation program that runs under Windows. With 20-sim one can simulate the behavior of dynamic systems, such as electrical, mechanical and hydraulic systems or any combination of these systems.

**Automod:** This software provides true to scale 3-D virtual reality animation, making simulation models easy to understand. It provides advanced features to allow users to simulate complex movement, such as kinematics and velocity of equipment such as robots, machine tools, transfer lines, special machinery and more. Automod provides the user with a set of expert-based material handling systems that have been developed with real-world experience in industrial automation. These include conveyors, power & free, asrs (automated storage and retrieval system) and bridge crane.

**Awesim:** Awesim provides a simulation engine focused on the production of model animations. Animations can be built graphically and the user can specify controls to build interactive simulations.

**Matlab:** This is an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high level programming language.

**Promodel:** This software is a simulation-based software for evaluating, planning or re-designing manufacturing, warehousing and logistics systems. Typical applications include implementation of lean manufacturing, cycle-time reduction, equipment investment decisions, throughput & capacity analysis, identifying and minimizing bottlenecks and resource allocation.

**Prosolvia:** This software brings reality to virtual manufacturing. It allows the user to view the product in a variety of views and situations. End user interaction and functional testing in the early concept stages of the product development process means that important design and manufacturing decisions can be made much earlier in the process. For the manufacturing process, staffing training for assembly and maintenance can be developed much earlier in the pre-production process.

**Quest:** This software provides the user with a graphics and visual analysis capability. It enables accurate modeling of conveyors, buffers, docks, asrs, etc.

**SDI supply chain:** This software provides a tool for studying the impact of changing demand, logistics decisions and production policies on key system performance measures. It is used to model the dynamics of a complete supply chain from source to user, along the entire plan, source, and make and deliver process.

**Simba:** Lanner's Simulation base application software enables the rapid development and component build of simulation embedded operational planning and scheduling applications for supply chains.

**Simplorer:** A simulator for industrial design, research projects and teaching purposes. External code can be embedded. Has its own language. It Allows for hierarchical structures.

**Witness (SDX):** This software offers discrete event simulation. This software has various facilities, which enable easy simulation of automotive manufacturing. Cycle times, breakdown modes and timings, setup modes and timings, buffer capacities, buffer dwell times, machine type etc. can be easily brought out as output together with routing information. It also includes material flow optimization. Effective material flow planning minimizes the high cost of moving materials and products from one machine to another.

### **1.3 Reason for Selecting Arena 7.0 Environment**

After careful consideration Arena 7.0 was selected as the software for carrying out the project. Following are the reasons justified for the same:

1. Animation Capability :
  - a. Arena 7.0 supports animation and the material flow through the system can be explained in a clear fashion to members of other teams who are not familiar with simulation.

- b. It would help in understanding the interface of simulation with Manufacturing, Transportation, Warehousing and Information Systems and clear the basic concepts in those fields.
- c. Animation acts as a debugging tool and helps to detect any errors made during building the basic blocks.

2. Syntax:

Arena 7.0 provides a user-friendly environment without involving any programming statements. The function of every "block" and "element" is hard-coded in advance, while developing the software.

3. Arena verification and debugging tools:

The in-built features in ARENA 7.0 helps to examine in detail the movement of entities through the system. It provides a summary of the action taken at each block through which an entity has passed. It also helps to step through the execution of the model by exercising complete control from the keyboard or mouse.

4. Customized reports:

The availability of customized reports at the end of each run helps in analyzing the results and take remedial steps to further improve the model.

5. Greater familiarity:

Arena environment is widely popular in US corporate and academic organizations.

6. Probability Distribution:

The "Input Analyzer" feature helps in generating probability distributions which

can be fed into the various blocks in the model for the random generation of entities.

## **Chapter 2**

### **2.1 AGV Introduction**

Automatic Guided Vehicles have been in existence since the 1950's. AGVs are driver-less industrial trucks, usually powered by electric motors and batteries. AGVs range in size from carrying small loads of a few kilograms up to loads over 100 tons. The working environment may vary from offices with carpet floor to harbor dockside areas.

Modern AGVs are computer-controlled vehicles with onboard microprocessors. Most AGV-systems also have system management computers, optimizing the AGV utilization, giving transport orders, tracking the material in transfer and directing the AGV traffic.

Automatic load handling is used in many AGV-systems. The AGV can pick up and drop off pallets or transfer loads automatically using fork attachments, conveyors, lift tops etc. depending on the type and size of the load units to handle.

The early AGVs were built as tuggers, towing trailers, or shaped as platform vehicles. The only function was to transport. AGVs today can be equipped with robot arms and grippers and perform robotic handling functions. AGVs are also used as storage machines equipped with forks, handling loads in storage racks up to 10 meters in height or more.

Several methods of guidance and navigation can be implemented. The early AGVs were tracking an inductive guide wire or an optical visible line, painted or made with tape on the floor. The inductive guide wire is still the most used guiding system for AGVs running on concrete floors, also for new installations.

Many companies and people have tried to invent the replacement for the inductive guide wire. In later years AGV guiding and navigation systems with laser scanners, microwave transponders, inertia gyros, ultrasonic sensors, embedded magnets, camera vision systems etc. have been launched.



**Figure 1** AGV controlled by floor-imbedded guidance wire

## **2.2 Major categories of AGVs:**

**Tow Type:** Used for pulling carts, trailers, dollies, etc. Typically range from 10,000 to 50,000 pounds towing capacity. Train lengths depend on capacity, space and trailer tracking performance.

**Unit Load Type:** Self-contained AGVs that carry products on their built-in load decks. Typically transport pallets, totes, rolls and racks, usually one or two at a time. Often incorporates automatic load transfers at pick and drop stations.

**Fork Type:** Utilizes a fork/mast lift mechanism for interfacing with loads at various elevations. Often loads are at floor or stand level and are retrieved and deposited automatically.

**Commercial/Office Type:** Smaller type units with capacities of less than 500 pounds. Designed to transport small totes or small loads in light manufacturing or clean environments. Versions are also used to distribute mail in offices.

**Heavy Carrier Type:** Designed to transport large or very heavy loads. Typically used in primary metal and paper industries. Capable of transporting dies, rolls, coils, ingots weighing in excess of 250,000 pounds.

## **Chapter 3**

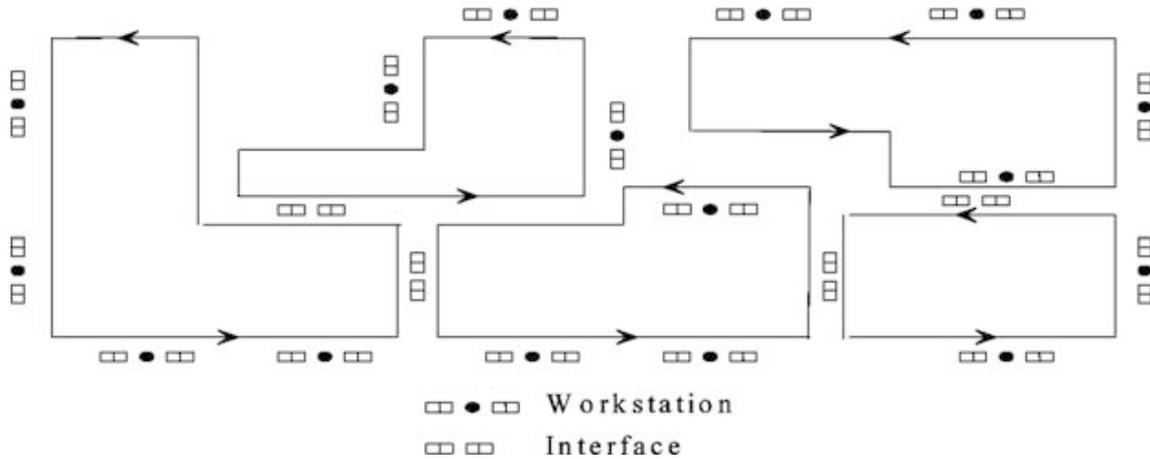
### **3.1 Simulation of AGV systems**

An automated guided vehicle (AGV) system is an advanced material handling system where driverless vehicles follow a guide-path and are controlled by an off-board computer or microprocessor. Unlike other, more conventional material handling equipment, an AGV is able to select its own route or path to reach its destination. Through its off-board controller or microprocessor, the vehicle is able to receive such dispatching commands as identification of the load, its destination, etc. In particular, because microprocessor technology has developed over the past decade, AGV systems are considered highly flexible material handling systems for the effective operation of such systems as flexible manufacturing systems (FMS) and flexible assembly systems. Several efforts to make AGV systems more flexible are underway with a view to eliminating the constraints which are imposed by a floor-imbedded guidance wire.

Two major concerns for AGV systems have been system design and control. The system design includes such issues as track layout and the types and number of vehicles required, while system control covers such issues as dispatching vehicles and part routing. The success of any AGV system is highly dependent on the quality of the system design and



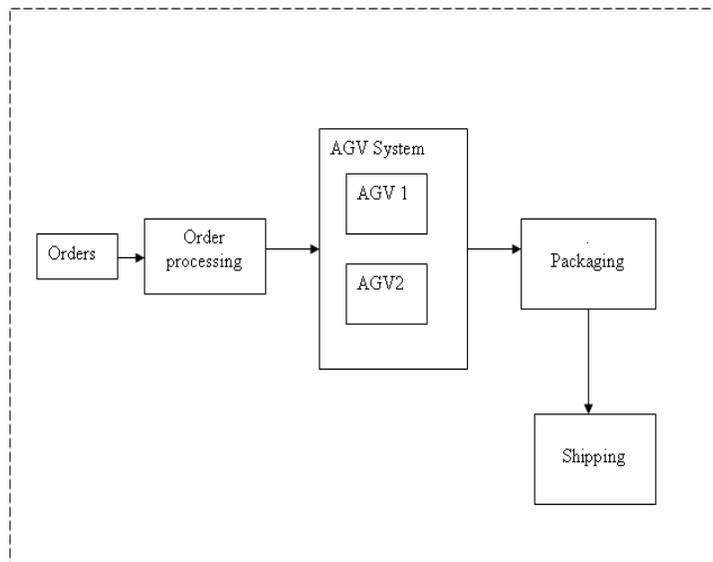
their inherent functional complexity and limitations. Also, in most dynamic manufacturing environments today, systems and processes are constantly changing. Simulation tools are required that can accurately model a system in detail, yet still be easy to use, and allow rapid model redevelopment to react quickly to system changes.



**Figure 2** Conventional and Tandem AGV System

### 3.2 The Warehouse System

The warehouse operation simulated in this dissertation is that of a system consisting of order processing unit, AGV system and packaging system.



**Figure 3** Block diagram of Warehouse system

The warehouse stores 2 types of valves- bronze and steel. These valves are stored in unpacked condition on the pallets. The system has automatic storage and retrieval system for picking up the order from the pallets. The order received is processed by the order processing unit. The order processing typically involves determining the number units of valves, type of valves and the product mix. The order data is transmitted to the AGV system. The AGV system decodes the data and determines the numbers of units to picked of each type of valve and computed the shortest possible path to pick the order. The AGV travels to the storage racks and the automatic storage and retrieval system loads the appropriate number of units of valves from the pallets on the AGV. The AGV carries this order to the packaging unit where the valves are boxed & labeled and the finished order is ready for shipment.

## **Chapter 4**

### **4.1 Simulation of the System**

#### **4.1.1 Arrival Rate**

Arrival-counting process is one of the important of all input models because simulations of industrial and service systems are typically driven by the arrivals of customers, orders, materials, information, etc. Renewal arrival-counting processes, in which the times between arrivals are i.i.d. random variables, are a standard feature of commercial simulation languages. A marginal distribution that is often chosen is the exponential distribution with mean  $1/\lambda$ , which implies that the renewal arrival process is a Poisson arrival process with constant arrival rate  $\lambda$  arrivals per unit time. The table below shows the inter-arrival data used for this project.

1	0.0	22	0.96	43	0.08
2	0.62	23	2.65	44	0.08
3	0.6	24	4.92	45	2.77
4	0.23	25	0.27	46	1.23
5	2.67	26	8.43	47	0.3
6	2.9	27	0.33	48	3.45
7	1.31	28	0.5	49	2.97
8	0.55	29	1.17	50	4.95
9	2.09	30	1.0	51	0.02
10	1.15	31	1.83	52	0.01
11	0.68	32	0.39	53	0.04
12	0.57	33	0.2	54	1.93
13	1.96	34	3.36	55	6.25
14	2.47	35	1.94	56	0.68
15	2.1	36	4.3	57	0.15
16	4.3	37	0.9	58	10.04
17	0.75	38	0.01	59	6.81
18	1.3	39	0.02	60	1.5
19	0.45	40	2.05	61	2.15
20	1.4	41	2.25	62	3.84
21	2.42	42	0.67	63	0.43

**Figure 4** Inter Arrival Data

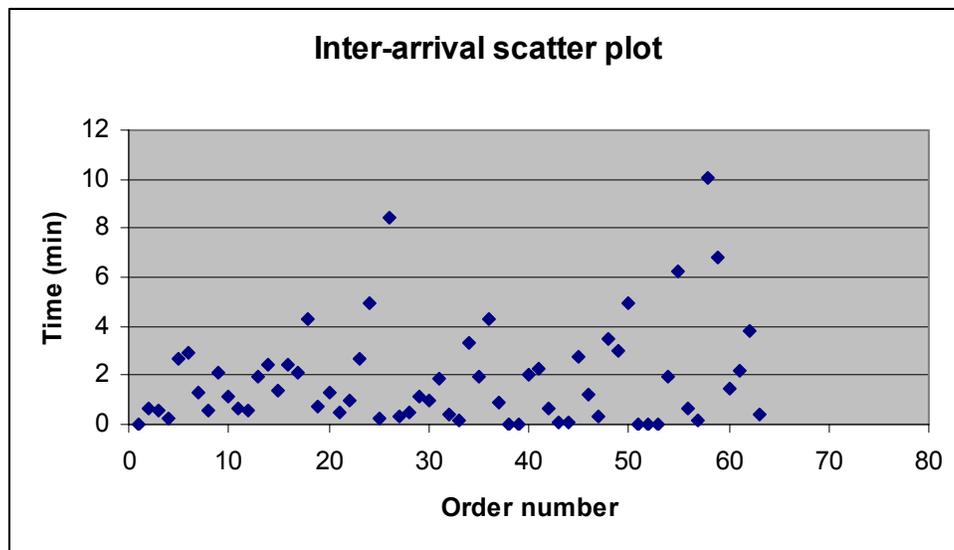
#### 4.1.2 Input modeling

A simulation model is composed of inputs and logic; the inputs represent the uncertainty or randomness in the system, while the logic determines how the system reacts to the uncertain elements. Simple input models, consisting of independent and identically distributed sequences of random variates from standard probability distributions, are included in every commercial simulation language.

Input modeling-choosing the representation-is often characterized as “picking a probability distribution.” And it may be that simple if the following approximations are reasonable:

- The input process can be represented as a sequence of independent random variables having a common (identical) distribution; in other words, a sequence of i.i.d. random variables.

- The common distribution is one of the standard families that are included in nearly all commercial simulation languages: beta, empirical, Erlang, exponential, gamma, lognormal, normal, Poisson, triangular, uniform or Weibull.
- Data are available from which to select and fit the distribution using methods such as maximum likelihood or moment matching.
- A standard distribution provides a good fit to the data, as verified by a visual inspection or a goodness-of-fit

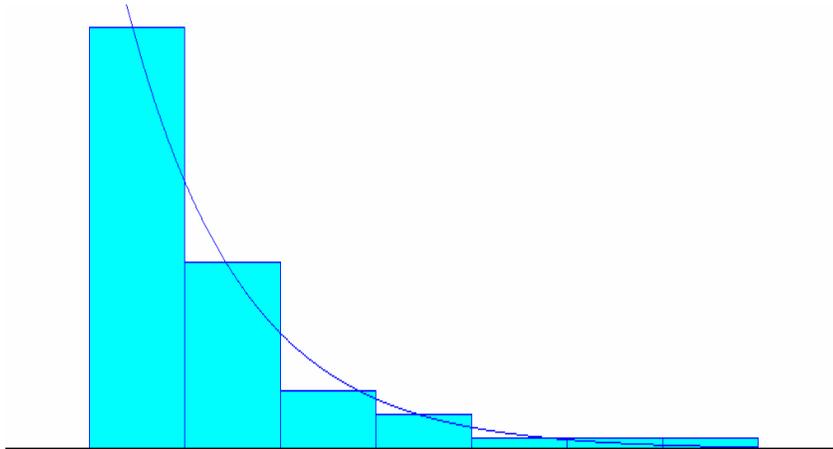


**Figure 5** Inter-arrival Scatter plot

The random numbers were generated using the function  $+1.5 * \text{NORMSINV}(\text{RAND}())$  for 63 data points. The generated data for 63 points was copied in separate text files. This file was called into the Input analyzer using (File/Datafile/UseExisting) option and thereafter 'fit all' test was performed on the data.

The data was arranged in order statistic and was checked for stationarity by drawing a scatter plot. The scatter plot was drawn as observation on x-axis and values (not ordered) on y-axis. Thereafter a histogram was plotted and the trend of the histogram was looked

at to identify and choose the best possible standard distribution that could be fitted to this data. In this case Exponential distribution best fits the data.



**Figure 6** Arena Input Analyzer plot

Distribution Summary

Distribution: Exponential

Expression:  $-0.001 + \text{EXPO}(0)$

Square Error: 0.001044

Chi Square Test

Number of intervals = 3

Degrees of freedom = 1

Test Statistic = 0.491

Corresponding p-value = 0.49

Kolmogorov-Smirnov Test

Test Statistic = 0.0833

Corresponding p-value > 0.15

#### Data Summary

Number of Data Points = 63

Min Data Value = 0

Max Data Value = 10

Sample Mean = 1.88

Sample Std Dev = 2.07

#### Histogram Summary

Histogram Range = -0.001 to 11

Number of Intervals = 7

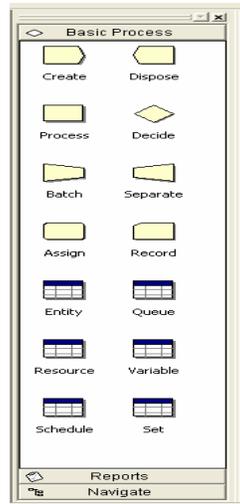
## Chapter 5

### 5.1 Simulation Concepts

#### 5.1.1 Basic Processes

The Arena model is built using modeling shapes, called modules, from the Basic Process panel to define the process. There are two types of modules on a panel: Flowchart modules and Data modules. Flowchart module shapes are placed in the model window and connected to form a flowchart, describing the logic of the process. Data modules are not placed in the model window. Instead, they are edited via a spreadsheet interface. Flowchart and Data modules are related via names for objects like Queues, Resources,

Entity types, Variables, etc. In the model window, there are two main regions. The model workspace contains all of the model graphics, including the process flowchart, animation, and other drawing elements. The lower, spreadsheet view displays model data, such as times, costs, and other parameters.



**Figure 7** Project Bar with basic processes

Clicking the module icon on the Project bar will display in the spreadsheet all modules of that type. A flowchart module's spreadsheet view displays the modules of that type for the active/currently displayed sub model view, but because data modules are global to the entire model, a data module's spreadsheet view displays all existing data modules of that type.

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Minimum	Value	Maximum	Report Statistics
1	AGY	Standard	Seize Delay Release	Medium(2)	2 rows	Triangular	Minutes	Value Added	3	5	6	<input checked="" type="checkbox"/>
2	Order processing Time	Standard	Delay	Medium(2)	0 rows	Triangular	Minutes	Value Added	6	10	12	<input checked="" type="checkbox"/>
3	Packaging	Standard	Delay	Medium(2)	0 rows	Triangular	Minutes	Value Added	5	7	9	<input checked="" type="checkbox"/>

**Figure 8** Spreadsheet view of the model

The model is initially displayed in split screen view. The top pane is used for placing modules to depict the flow of the system, and the bottom pane is the spreadsheet used to enter data. The amount of space displayed for either pane can be changed by moving the splitter bar up or down. In full view, Arena will display the appropriate pane for the module that is selected. If the flowchart module icon in the panel is clicked, the model workspace is displayed; and if the data module icon is clicked, the spreadsheet section is displayed. In full view the flowchart modules cannot be edited via the spreadsheet since each time the flowchart module icon in the panel is clicked, the model workspace is activated.

When the 'Go' is hit to run the model, Arena automatically hides the spreadsheet portion of the model. The spreadsheet can be displayed during the run by either toggling the View/Split Screen option, or moving the splitter bar to a desired position. When the run is stopped, the view will return to its previous setting.

### **5.1.2 Entities and Attributes**

In a simulation model, entities represent the objects moving through the system. Each entity has its own characteristics, referred to as attributes. You can define as many attributes as you need for the entities in your system. Each individual entity in the system has its own values of these attributes; these may be assigned at the various processes it encounters.

For example, you may want to have an attribute named "Length" to define the entity size, and one named "Priority" to establish the importance of the entity in relation to some

activities. When the entity performs some process (such as loading onto a conveyor) its own particular attribute value (i.e., its size as defined by its "Length" attribute value) dictates the space it requires. In this model the entities are bronze valves order steel valve order.

### **5.1.3 Queues**

The primary purpose of a queue is to provide a waiting space for entities whose movement through the model has been suspended due to the system status (e.g., a busy resource). Queues are passive in nature; entities enter the queue and are removed from it based upon the change in state of the system element associated with the queue (e.g., a resource).

There are two types of queues used by Arena. Individual queues have a symbolic name, a ranking rule, and a specific capacity. Entities in these queues may be displayed in the animation; statistics may be collected on them; they may be ranked using a flexible ranking rule mechanism; they may be collected into sets and, when used with resources, they may be shared among modules. Since the action logic in the AGV process in the model is 'Seize, Delay, Release' AGV.que is formed in the system. Internal queues provide a basic first-in, first-out container for entities at a particular activity (module), but do not provide animation, statistics or ranking mechanisms.

### **5.1.4 Resources**

Resources are stationary elements of a system that can be allocated to entities. They have a specified capacity (at any point in time) and a set of states (e.g., busy, idle, inactive, or

failed) that they transition between during a simulation run. Resources may be used to represent people, machines, or even space in a storage area.

The terminology associated with resources is as follows: when an entity requires a resource, it seizes the resource; and when an entity no longer requires a resource, the entity releases it so that it is available to be seized by other entities. A resource has an associated queue to hold entities that try to seize the resource when it is unavailable.

The capacity of a resource limits the number of entities that may seize it at any point in time. For instance, a table that can seat four customers could be represented by a resource called TableA having a capacity of four. An entity that seizes a resource is referred to as seizing a unit from its total capacity. Entities can seize and release multiple units of capacity. They can also seize and release fractions of a resource unit. For example, an arriving entity representing a family might seize all four units of TableA; or, two entities might each seize two units of the resource; or, an entity might seize .25 units of the resource.

Resource capacities may be changed over time by defining a resource schedule that adds and removes units of the resource to represent shifts, seasonal staffing patterns, etc. Downtimes and failures may also be associated with resources to capture planned and unplanned activities, such as breaks and maintenance.

The multi-capacity resource is represented in the model as a single resource for purposes of statistics collection (e.g., statistics represent the entire resource TableA rather than the four individual seats) and animation. If specific statistics or separate animation pictures

of each capacity unit are required, define an individual resource for each unit and collect the resources into a set.

Resources are depicted in the animation by a stationary set of pictures representing the states of the resource (idle, busy, etc.) The default pictures can be customized to better represent the resources in your system.

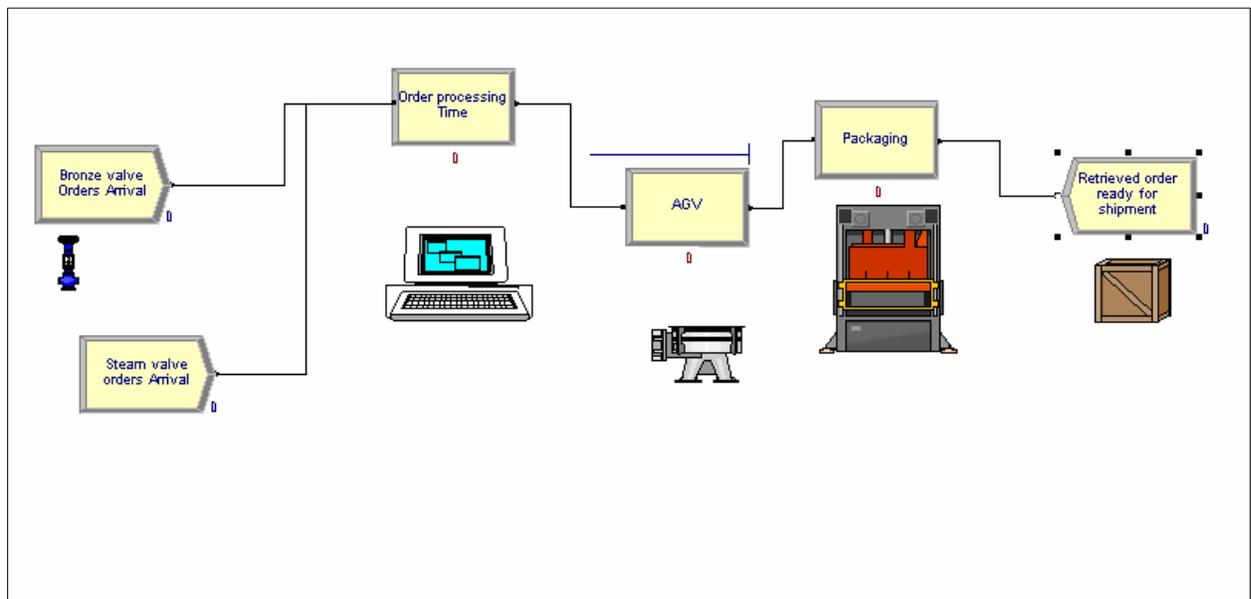
### **5.1.5 Statistics**

Arena allows you to specify whether or not simulation statistics are to be collected for a particular element (e.g., queues, resources, etc.) These statistics are recorded during the simulation run and are displayed in a report that you may view at any time during the run or after the run is completed. Statistics collected by Arena fit into one of five categories: Time-persistent record time-weighted statistics (e.g., queue lengths and resource utilizations). Tallies record observational data (e.g., the time entities spent in the system). Counters count the number of observed occurrences of some event (e.g., number of items leaving the system). Frequencies collect category-based statistics on a system variable (e.g., proportion of time a resource spends in each of its possible states). Finally, outputs report the final value of some system or statistical value (e.g., inventory level at end of run).

## Chapter 6

### 6.1 Arena Model:

Arena provides an intuitive, flowchart-style environment for building an as-is model of the warehouse operations process. Arena model created in this dissertation consists of two Create, three Processes and one Dispose flowchart modules.



**Figure 9** Arena Model for the warehouse system

The two Create modules are called 'Bronze valve orders Arrival' and 'Steel Valve orders Arrival'. The three processes are called 'Order Processing Time', 'AGV' and 'Packaging'. The Dispose module is called 'Retrieved order ready for shipment'. Different pictures from the media library are used to represent each flowchart module. The flowchart module is added by dragging the basic processes from the Project Bar into the flowchart view of the model window. All the dynamic processes are defined with set values and conditions by double clicking on the basic processes blocks and validating the

appropriate fields. Arena keeps internal lists of different kinds of names. It presents existing lists to you where appropriate. It helps you remember names and protects you from typos. All the names you made up in a model must be unique across the model, even across different types of modules. In Arena all the dynamic processes have ‘nodes’ through which the entities flow. All the nodes are typically connected and finally the entity terminates in the Dispose module. The dynamic processes are connected using the ‘connect’ button on the standard tool bar.

## 6.2 Model components

### 6.2.1 Order Creation

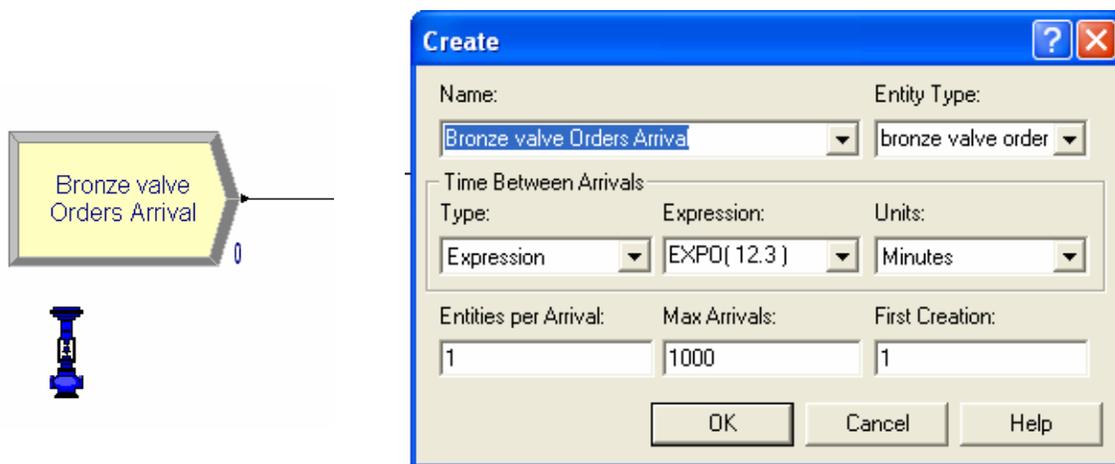


Figure 10 ‘Bronze Valve Orders Arrival’-Create

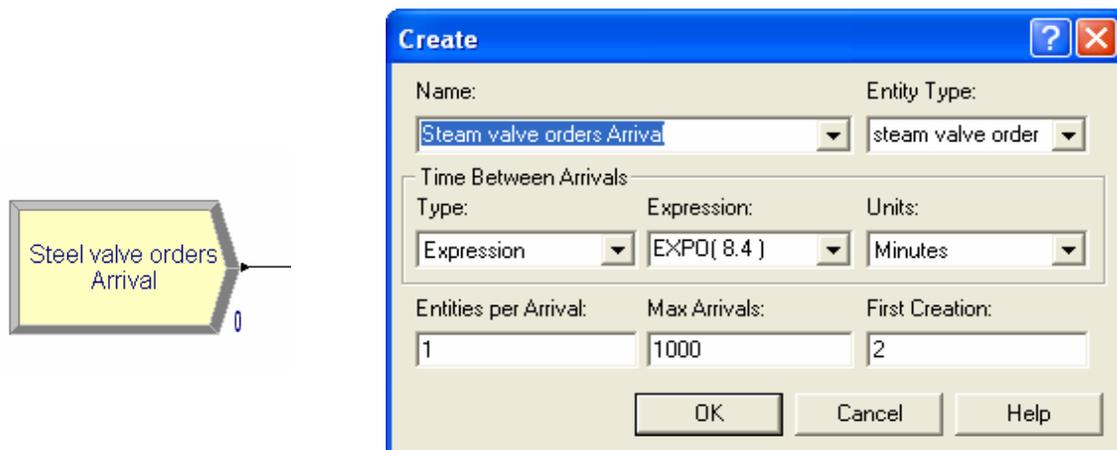


Figure 11 ‘Steel Valve Orders Arrival’-Create

The create block in Arena is used to generate the entities. This model uses two create blocks to generate the two different types of orders. The first create block generates entity which represents a bronze valve order. The name of the entity can be defined in the 'Entity Type' text field. In this case the entity is called 'bronze valve order'. The second create block generates the steel valve orders. The entities are recognized by the same attribute or name until they are disposed in the dispose box. In this model two different entities are flowing through the model and Arena distinguishes the two types of entities viz. steel valve order and bronze valve order by their name. The create block can be defined in the 'Name' text-field. The name of the create block and the name of the entity that it creates are not correlated. The time between arrivals, in other words the creation of new entity, is governed by any one of the four formulae in the 'type' section of create block. The four formulae are random, schedule, constant and expression. The 'random' option generates the entity randomly and does not follow any type of distribution. For this option the input data modeling is not required to determine the type of distribution followed by the data. The simulation of this type of input involves high degree of randomness. In the 'schedule' and 'constant' option the randomness is relatively less and the user defines the input pattern. The 'schedule' option requires the user to specify the arrival patterns for length of time. The 'constant' option creates entities periodically with fixed time interval between two creations. For the 'expression' option Arena provides quite a few distributions. The various types of expressions available to randomly create an entity are Exponential, Normal, Triangular, Uniform, Erlang, Beta, Gamma, Johnson, Constant, Lognormal, Weibull and Poisson. The 'expression' option is most common in

the practical simulation studies in the industrial environment. The type of ‘expression’ to be chosen depends on the input modeling. After running the data through the Arena Input Analyzer the type of distribution is determined. This distribution is selected for the ‘expression’. In this model after the input analysis the Input Analyzer defined the data to follow exponential distribution. Hence the create block in the model uses the exponential distribution. The units for time between arrivals are selected as minutes. The other parameters that need to be specified in the create block are ‘entities per arrival’, ‘maximum arrival’ and ‘first creation’. First creation is the time when first entity will be created. Entities per arrival field is used to specify the number of entities created at a time. The maximum arrival is the total number of entities created during the period of simulation. The ‘replication length’ on replication parameters dialog and the ‘maximum arrival’ decide the length of the simulation. The simulation stops as soon as the replication length, in other words the maximum time for one simulation cycle, is reached or maximum number of entities are created.

### 6.2.2 Order Processing Time

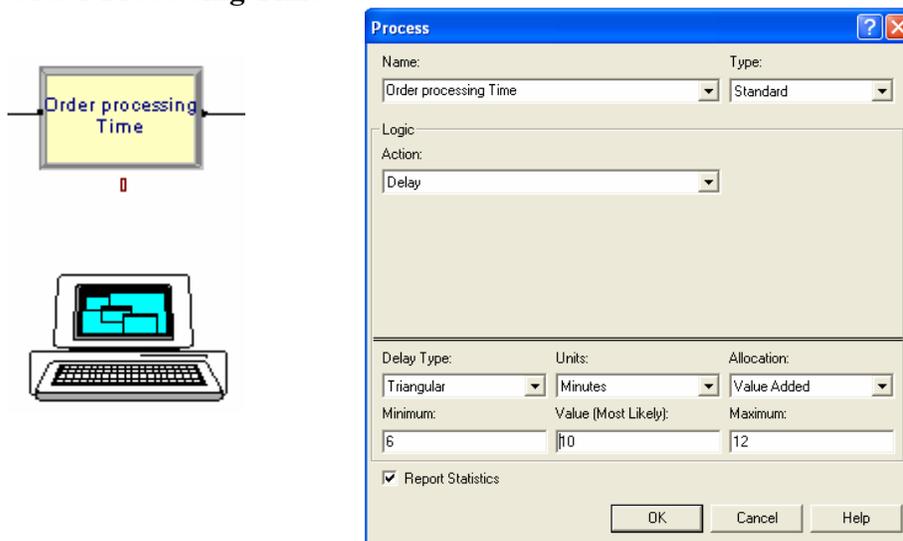
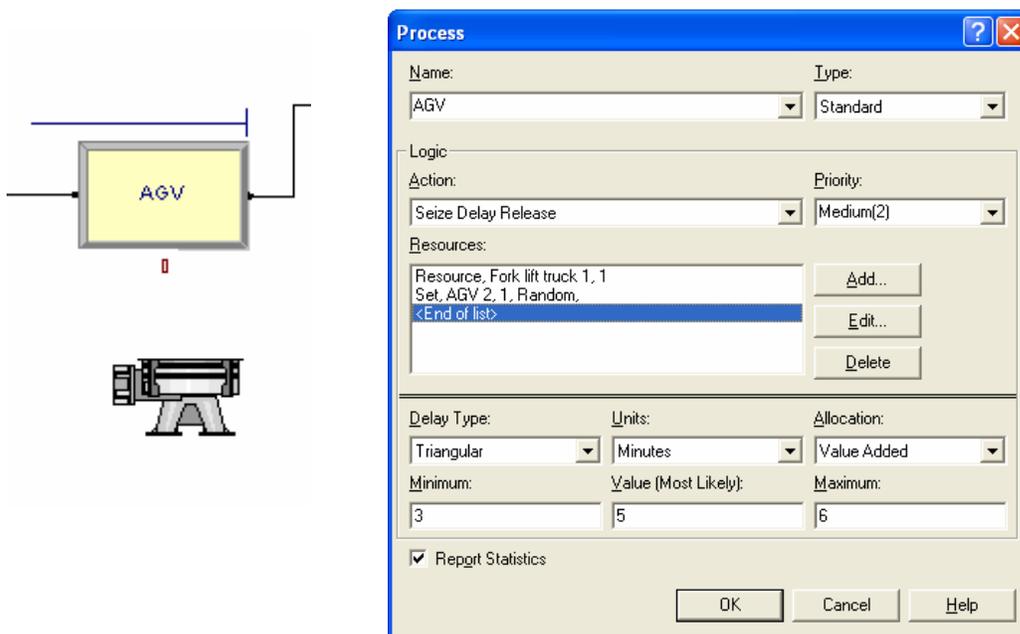


Figure 12 ‘Order Processing Time’ - Process

The 'order processing time' flowchart module represents the processing station and the entity delay time. The type of process is picked as standard. Order processing is supposed to be done by a computer and hence the logic action is selected as delay. The delay type is chosen as Triangular with processing time of minimum 6 min, most likely 10 min and maximum 12 min. The Report Statistics check box is ticked at the bottom to get the utilization, queue lengths and queue waiting times.

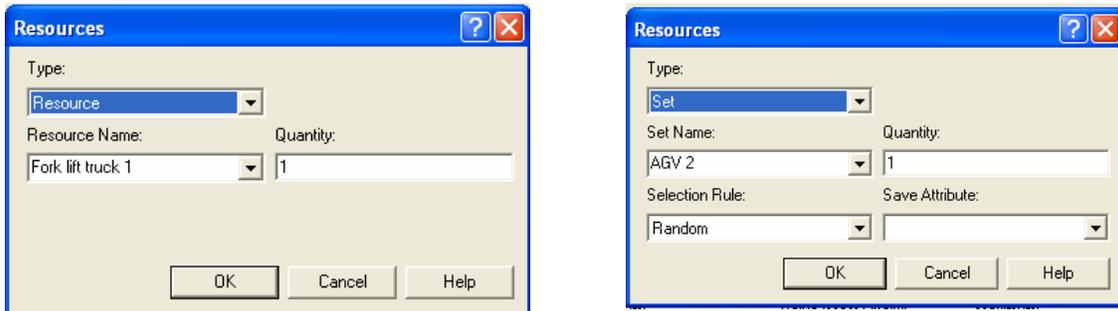
### 6.2.3 AGV



**Figure 12** 'AGV'- Process

The 'AGV' flowchart module represents the resources- 'Fork lift truck1' & 'AGV2' and the entity delay time. The type of process is picked as standard. In the logic area the action is chosen as 'Seize Delay Release'. The order or entity seizes the resources in the AGV system and delays itself. The delay is due to the processing time required by the AGV system to pick up the order of valves from pallets using the A-R systems and then the entity releases itself. The delay type is chosen as Triangular with processing time of minimum 3 min, most likely 5 min and maximum 6 min. Since the logic in this flowchart

module is Seize Delay Release, a queue is formed in the system. The name of the queue is AGV.queue. The Report Statistics check box is ticked at the bottom to get the utilization, queue lengths and queue waiting times. The AGV system consists of two resources shown below in the figure. A new resource is added by clicking the Add button

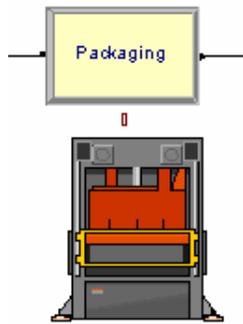


**Figure 13** Resources and Set

under the Resources area. The first resource is named 'Fork Lift Truck 1' and its quantity is 1. The second resource is a Set. A Set is a collection of resources that are according to the selection rule. The various selection rules available are cyclical, random, preferred order, specific member, largest remaining capacity, smallest number busy. In this model the selection rule is chosen as Random as the fork lift truck or AGV is chosen randomly depending upon the availability.

#### **6.2.4 Packaging**

The entity releases itself from the AGV system and reaches the packaging section. The packaging is performed by packing machines and the Action under the Logic area is chosen as Delay. The 'Packaging' flowchart module represents the Packaging and the entity delay time. The type of process is picked as standard.



**Figure 14** Packaging' - Process

The delay type is chosen as Triangular with processing time of minimum 5min, most likely 7 min and maximum 9 min. The Report Statistics check box is ticked at the bottom to get the summary statistics.

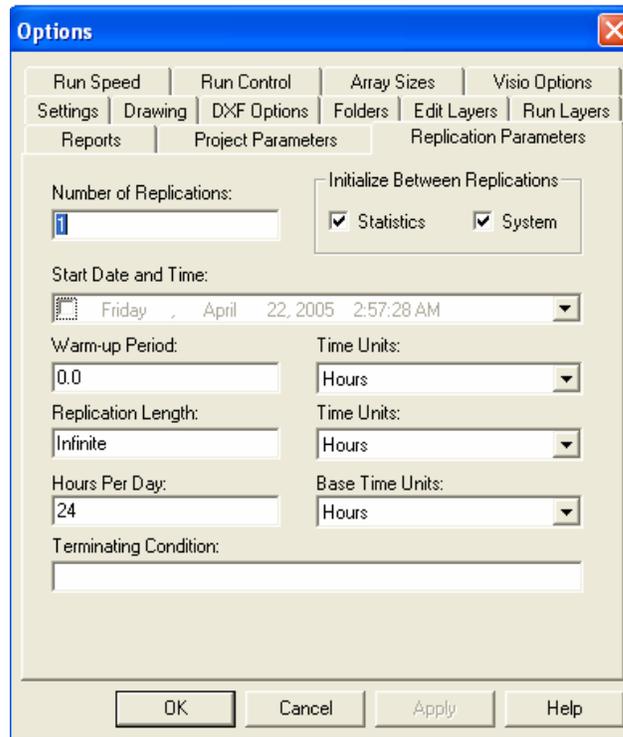
### 6.2.5 Retrieved order



**Figure 15** 'Retrieved order ready for shipment'-Dispose

All the entities created by the 'bronze valve order arrival' Create and 'steel valve order arrival' Create modules are collected in the dispose block. The recorded entity statistics give the time spent by entity in the system, waiting times, queue lengths etc.

### 6.3 Setting the Run Conditions

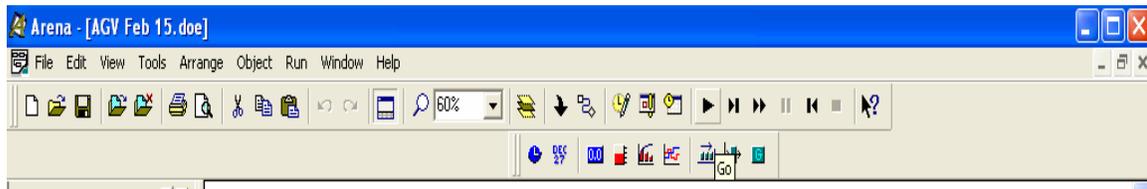


**Figure 16** Run Setup Dialog

In the Run/Setup menu dialog the Project Parameters – Title, name, output statistics are selected. The output statistics collected in this model are entities, queues and resources. Replication Parameters tab specifies – Number of Replications, Length of Replication (and Time Units), Base Time Units (output measures, internal computations), Warm-up Period (when statistics are cleared) and Terminating Condition (complex stopping rules) Other three tabs specify animation speed, run conditions, and reporting preferences

Part of modeling is to specify the termination of the program. Arena has no default termination. If termination is not specified Arena will usually keep running forever. In this model the in each Create block the Maximum Arrivals is limited to 1000. So the simulation for one replication is stopped as soon as the maximum number is reached.

## 6.4 Running the model



**Figure 17** Play Button on the Standard Tool Bar

The Arena model is run by clicking the Play button the standard tool bar. The internal compiler first enters the check mode and looks for any errors in the code due to incorrect specification of parameters in any of the flowchart modules. If the compiler does not find any errors it enters the run mode. The user cannot edit the model when Arena is in the run mode. The speed of the simulation could be speed up by '>' key on the keyboard or slowed down using the '<' key. When done Arena generates user specified summary reports.

## Chapter 7

### 7.1 Siman Code

```
Model statements for module: Create 1
;
4$      CREATE,      1,NSEXPO(Schedule 1),bronze valve order:NSEXPO(Schedule
1),1000:NEXT(5$);
5$      ASSIGN:      Bronze valve Orders Arrival.NumberOut=Bronze valve Orders
Arrival.NumberOut + 1:NEXT(2$);
;
;
;      Model statements for module: Process 2
;
2$      ASSIGN:      Order processing Time.NumberIn=Order processing
Time.NumberIn + 1:
                Order processing Time.WIP=Order processing Time.WIP+1;
9$      DELAY:      Triangular(6,10,12),,VA;
```

```

56$    ASSIGN:    Order processing Time.NumberOut=Order processing
Time.NumberOut + 1:
          Order processing Time.WIP=Order processing Time.WIP-
1:NEXT(0$);
;
;
;    Model statements for module: Process 1
;
0$    ASSIGN:    AGV.NumberIn=AGV.NumberIn + 1:
          AGV.WIP=AGV.WIP+1;
62$    QUEUE,    AGV.Queue;
61$    SEIZE,    2,VA:
          Fork lift truck 1,1:
          SELECT(AGV 2,RAN, ),1:NEXT(60$);

60$    DELAY:    Triangular(3,5,6),,VA;
59$    RELEASE:  Fork lift truck 1,1:
          SELECT(AGV 2,LAST),1;
107$   ASSIGN:    AGV.NumberOut=AGV.NumberOut + 1:
          AGV.WIP=AGV.WIP-1:NEXT(3$);

;
;
;    Model statements for module: Process 3
;
3$    ASSIGN:    Packaging.NumberIn=Packaging.NumberIn + 1:
          Packaging.WIP=Packaging.WIP+1;
111$   DELAY:    Triangular(5,7,9),,VA;
158$   ASSIGN:    Packaging.NumberOut=Packaging.NumberOut + 1:
          Packaging.WIP=Packaging.WIP-1:NEXT(1$);

;
;
;    Model statements for module: Dispose 1
;
1$    ASSIGN:    Retrieved order ready for shipment.NumberOut=Retrieved order
ready for shipment.NumberOut + 1;
161$   DISPOSE:  Yes;

;
;
;    Model statements for module: Create 2
;

```

162\$ CREATE, 1,MinutesToBaseTime(2),steam valve  
order:MinutesToBaseTime(EXPO( 8.4 )),1000:NEXT(163\$);

163\$ ASSIGN: Steam valve orders Arrival.NumberOut=Steam valve orders  
Arrival.NumberOut + 1:NEXT(2\$);

PROJECT, "AGV ","Ameya " ,,,No,Yes,Yes,Yes,No,No,No,No,No;

SCHEDULES: Schedule  
1,TYPE(Arrival),FORMAT(Duration),FACTOR(1.0),UNITS(Hours);

VARIABLES: Steam valve orders  
Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
Order processing Time.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
AGV.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
Packaging.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):  
Bronze valve Orders  
Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
AGV.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
AGV.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Retrieved order ready for  
shipment.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
Packaging.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):  
Order processing Time.WIP,CLEAR(System),CATEGORY("Exclude-  
Exclude"):  
Packaging.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):  
Order processing  
Time.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");

QUEUES: AGV.Queue,FIFO,,AUTOSTATS(Yes,,);

PICTURES: Picture.Airplane:  
Picture.Green Ball:  
Picture.Blue Page:  
Picture.Telephone:  
Picture.Blue Ball:  
Picture.Yellow Page:  
Picture.EMail:  
Picture.Yellow Ball:  
Picture.Bike:  
Picture.Report:  
Picture.Van:  
Picture.Widgets:  
Picture.Envelope:

Picture.Fax:  
Picture.Truck:  
Picture.Letter:  
Picture.Box:  
Picture.Woman:  
Picture.Package:  
Picture.Man:  
Picture.Diskette:  
Picture.Boat:  
Picture.Red Page:  
Picture.Green Page:  
Picture.Red Ball;

RESOURCES: Resource

1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):

Fork lift truck

1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):

Fork lift truck

2,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,);

REPLICATE,

1,,HoursToBaseTime(24),Yes,Yes,MinutesToBaseTime(1.0),,,24,Minutes,No,No,,Yes;

ENTITIES: steam valve

order,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,):

bronze valve order,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,);

SETS: Set 1,Fork lift truck 1:

Set 2,Fork lift truck 2:

AGV 1,Resource 1:

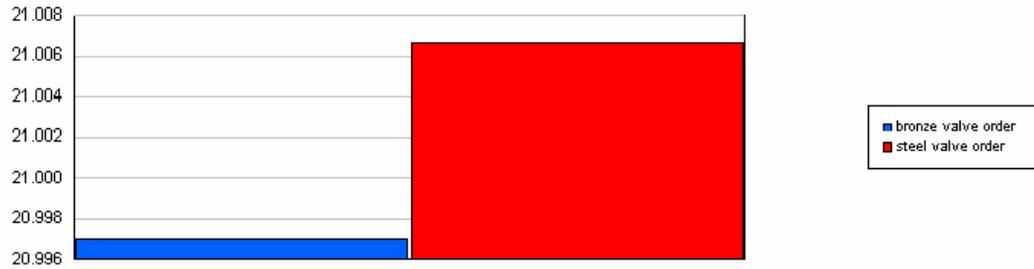
AGV 2,Resource 1;

## Chapter 8 Arena Output Results

### 8.1 Entity:

#### 8.1.1 Value Added Time

VA Time	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	20.9970	(Insufficient)	16.5350	24.3327
steel valve order	21.0067	(Insufficient)	17.4078	26.0277

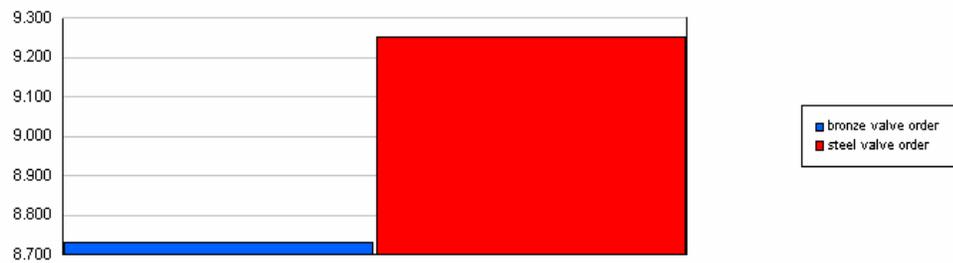


#### 8.1.1 Non Value Added Time

#### 8.1.2 Wait Time

NVA Time	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	0.00	(Insufficient)	0.00	0.00
steel valve order	0.00	(Insufficient)	0.00	0.00

Wait Time	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	8.7328	(Insufficient)	0.00	35.8957
steel valve order	9.2509	(Insufficient)	0.00	33.7894



### 8.1.4 Transfer time

### 8.1.5 Other Time

### 8.1.6 Total Time

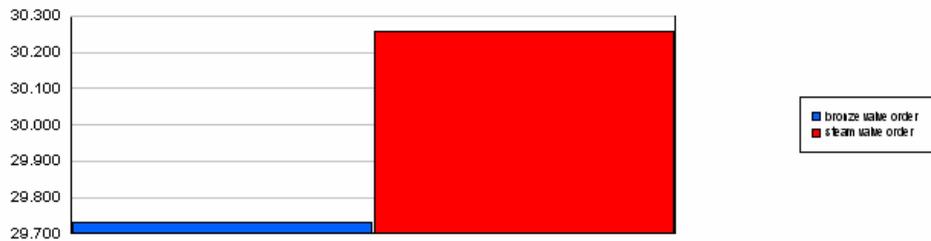
Transfer Time				
	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	0.00	(Insufficient)	0.00	0.00
steam valve order	0.00	(Insufficient)	0.00	0.00

Other Time				
	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	0.00	(Insufficient)	0.00	0.00
steam valve order	0.00	(Insufficient)	0.00	0.00

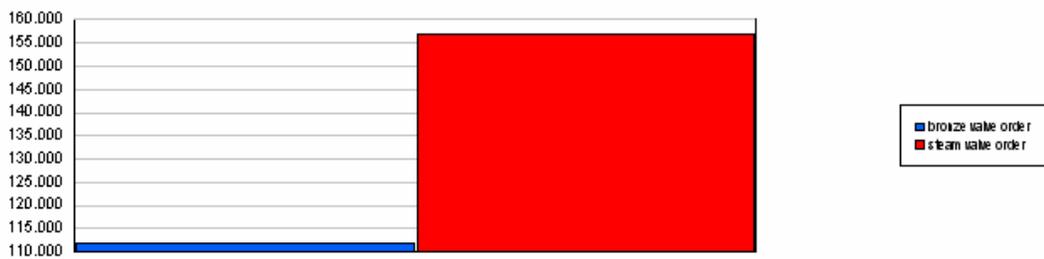
  

Total Time				
	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	29.7299	(Insufficient)	18.2504	56.2607
steam valve order	30.2575	(Insufficient)	18.1435	55.9718



### 8.1.7 Number in

Number In	
	Value
bronze valve order	112.00
steam valve order	157.00

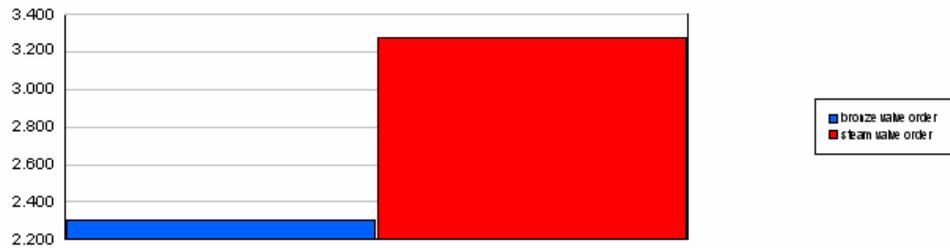


### 8.1.8 Number out

### 8.1.9 WIP

Number Out	Value
bronze valve order	111.00
steam valve order	153.00

WIP	Average	Half Width	Minimum Value	Maximum Value
bronze valve order	2.3030	(Insufficient)	0.00	7.0000
steam valve order	3.2746	(Insufficient)	0.00	10.0000



### 8.1.10 Waiting time

### 8.1.11 Number Waiting

Waiting Time	Average	Half Width	Minimum Value	Maximum Value
AGV.Queue	9.0700	(Insufficient)	0.00	35.8957

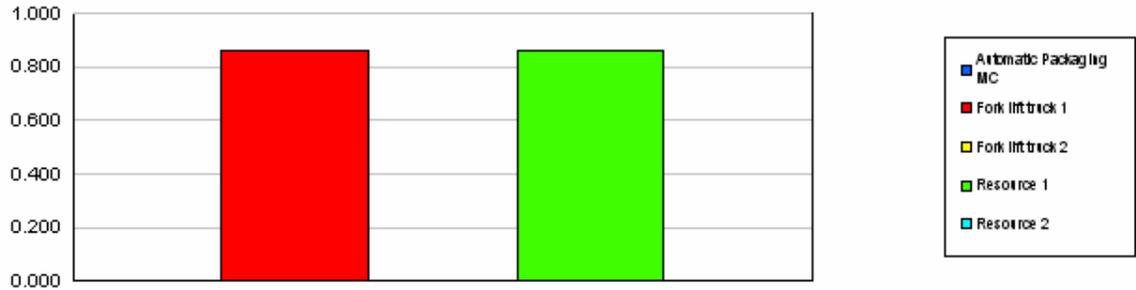
#### Other

Number Waiting	Average	Half Width	Minimum Value	Maximum Value
AGV.Queue	1.6884	(Correlated)	0.00	8.0000

## 8.2 Resources:

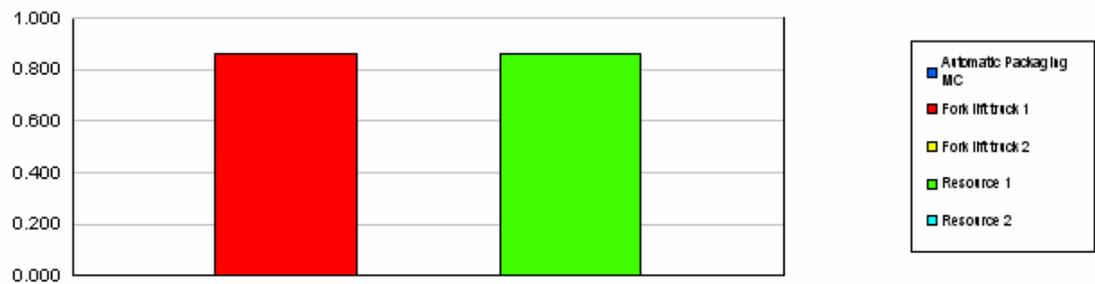
### 8.2.1 Instantaneous utilization

Instantaneous Utilization				
	Average	Half Width	Minimum Value	Maximum Value
Automatic Packaging MC	0.00	(Insufficient)	0.00	0.00
Fork lift truck 1	0.8611	(Insufficient)	0.00	1.0000
Fork lift truck 2	0.00	(Insufficient)	0.00	0.00
Resource 1	0.8611	(Insufficient)	0.00	1.0000
Resource 2	0.00	(Insufficient)	0.00	0.00



### 8.2.1 Number busy

Number Busy				
	Average	Half Width	Minimum Value	Maximum Value
Automatic Packaging MC	0.00	(Insufficient)	0.00	0.00
Fork lift truck 1	0.8611	(Insufficient)	0.00	1.0000
Fork lift truck 2	0.00	(Insufficient)	0.00	0.00
Resource 1	0.8611	(Insufficient)	0.00	1.0000
Resource 2	0.00	(Insufficient)	0.00	0.00



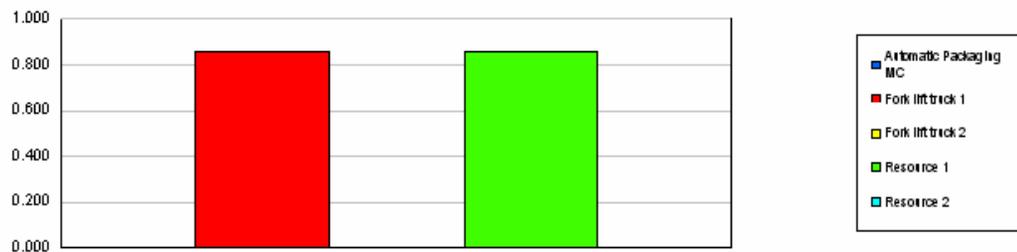
### 8.2.3 Number Scheduled

Number Scheduled	Average	Half Width	Minimum Value	Maximum Value
Automatic Packaging MC	1.0000	(Insufficient)	1.0000	1.0000
Fork lift truck 1	1.0000	(Insufficient)	1.0000	1.0000
Fork lift truck 2	1.0000	(Insufficient)	1.0000	1.0000
Resource 1	1.0000	(Insufficient)	1.0000	1.0000
Resource 2	1.0000	(Insufficient)	1.0000	1.0000



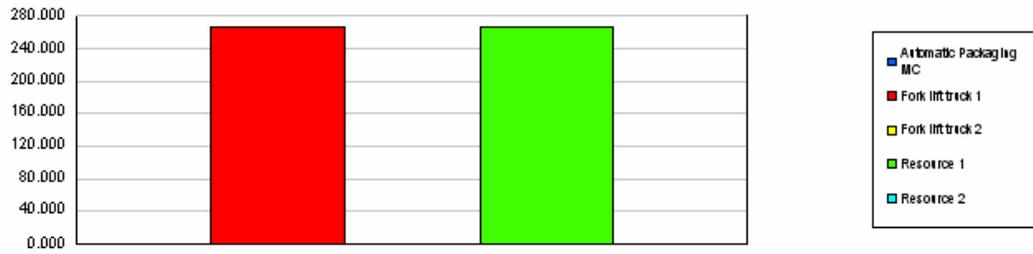
### 8.2.4 Scheduled utilization

Scheduled Utilization	Value
Automatic Packaging MC	0.00
Fork lift truck 1	0.8611
Fork lift truck 2	0.00
Resource 1	0.8611
Resource 2	0.00



## 8.2.5 Total number seized

Total Number Seized	Value
Automatic Packaging MC	0.00
Fork lift truck 1	266.00
Fork lift truck 2	0.00
Resource 1	266.00
Resource 2	0.00



## **Chapter 9**

### **9.1 Summary and Conclusions**

Arena offers numerous features to simulate discrete event systems. The enhance features in Arena allow to model the system for design, performance analysis and planning. An Arena-based warehouse environment has been described in this paper. Simulation is a very effective way of evaluating different scenarios in such an environment. Arena flowchart module was created using the Basic Processes. The input was modeled using Arena Input Analyzer. Based on the input distribution the system was simulated for one replication. The output results indicate the performance of the warehouse operations in various aspects viz. value added time, utilization, wait time, queue lengths, WIP etc. These parameters are vital in analyzing the operational efficiency of a system.

Future development for this project would be to simulate the system for larger number of replications. Arena Process Analyzer (PAN) could be used to compare the various alternatives and the Arena OptQuest could be used to search the optimal alternative. Additionally, the simulation can also be used to predict future system performance based on current decisions by running in fast mode from the current physical system state. The results of the fast mode analysis can be used by the decision making functions.

## 9.2 References

- [1] Sadowski, D., Sadowski, R., and D. Kelton, *Simulation with Arena*, McGraw Hill, 2004.
- [2] Knill, B., *Logistics Pros are Learning IT*, Material Handling Management, Vol. 55: No. 1, Jan 2000.
- [3] D.S. Hira and P.K Gupta, *Operations Research*, Dhanpatrai & Sons, 1998, 1999.
- [4] Grajo E., Gunal A., Sathyadev D., and Ulgen O., *A Uniform Methodology for Discrete-Event and Robotic Simulation*, Production Modeling Corporation, Dearborn, MI
- [5] Versteegt C., and Verbraeck A. *Extended Use Simulation Evaluating Real Time Control Systems of AGVS and Automated Material Handling Systems* Delft University of Technology The Netherlands
- [6] Browne, J., and K. Rathmill, *Simulation in Manufacturing*, IFS Publications/Springer-Verlag, 1998.
- [7] Hurrion, R. D., *Simulation Applications in Manufacturing*, IFS Publications Ltd., UK, 1996.
- [8] Avni, T., *Simulation Modeling Primer*, IIE Solutions, September 1999, 38-41.
- [9] Haddock J., and Voss P., *Simulation Based Performance Analysis of a Robotic Tested Control Architecture*, NASA Grant NAG-1-1469, June 1995
- [10] Choi, H.G., Kwon, H.J., Lee, J., *Traditional and Tandem AGV system layouts: A Simulation study*, Simulation, 63(1994)85-93.
- [11] Rocha R.P., and Carvalho A..S., *Stochastic and Colored Petri Nets for Modeling and Designing of AGV Systems* Praxis XXI/BD/18518/98
- [12] Buzacott, J. A., Shanthikumar, J.,G., *Stochastic Models of Manufacturing Systems*, Prentice Hall, 1993
- [13] Banks, J. and R. Gibson, *Don't Simulate When 10 Reasons for Determining When Simulation is Not Appropriate*, IIE Solutions, September 1997, 30-2
- [14] Banks, J., Ed., *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, New York: Wiley and Engineering & Management Press, 1998