



MASTER IN INFORMATION SYSTEMS (M.I.S.)

Thesis

**Developing simulation tools in Operations Management
(Ανάπτυξη εργαλείων προσομοίωσης στη Διοίκηση Λειτουργιών)**

Papanikolaou A. Maria

Supervisor: Andreas C. Georgiou

Thessaloniki, January 2011

Contents

CONTENTS	2
INTRODUCTION	4
1. PRELIMINARIES.....	4
2. FLEXIBLE MANUFACTURING SYSTEM (FMS)	5
2.1. A successful FMS implementation	6
3. AUTOMATIC GUIDED VEHICLES (AGVS).....	7
3.1. AGVs yesterday and today	8
3.2. Advantages	9
3.3. Disadvantages.....	10
4. SETTING UP AN AGV SYSTEM.....	11
4.1. Design of an AGV system	12
4.2. Control of an AGV system	12
4.3. Vehicle dispatching.....	13
4.4. Vehicle Scheduling.....	13
a. Requirements generation (open shop versus closed shop)	14
b. Processing complexity	14
c. Scheduling criteria	15
4.4.1. Scheduling rules	15
4.4.2. Some general comments	19
4.5. Vehicle Routing.....	20
a. Algorithms for general path topology	21
b. Path optimization.....	22
c. Algorithms for specific path topologies	23
d. Dedicated scheduling algorithm.....	24
5. MODELING AND SIMULATION OF AGV SYSTEMS	25
6. THE MODELS	27
a. Least Idle Vehicle (Least IV)	27
i. About the Least IV model	27
ii. Running the Least IV Simulation.....	29
b. Least Cumulative Idle Vehicle (Least CIV).....	35
i. About the Least CIV model	35

ii.	Running the Least CIV Simulation.....	37
c.	Longest Idle Vehicle (Longest IV).....	41
i.	About the Longest IV model.....	41
ii.	Running the Longest IV Simulation	42
d.	Longest Cumulative Idle Vehicle (Longest CIV)	47
i.	About the Longest CIV model.....	47
ii.	Running the Longest CIV model	48
e.	Least Utilized Vehicle (Least UV)	52
i.	About the Least UV model	52
ii.	Running the Least UV model.....	55
f.	Nearest Vehicle (NV)	58
i.	About the NV (distance) model.....	59
ii.	Running the NV (distance) model	62
iii.	About the NV (speed) model.....	65
iv.	Running the NV (speed) model.....	69
g.	Farthest Vehicle (FV).....	73
i.	About the FV (distance) model	73
ii.	Running the FV (distance) model.....	76
iii.	About the FV (speed) model	79
iv.	Running the FV (speed) model.....	83
h.	First In First Out Vehicle (FIFO).....	87
i.	About the FIFO model	87
ii.	Running the FIFO model.....	88
i.	Last In First Out Vehicle (LIFO)	91
i.	About the LIFO model	91
ii.	Running the LIFO model.....	92
j.	Random Vehicle (RV).....	94
i.	About the RV model.....	94
ii.	Running the RV model.....	96
7.	CONCLUSION	100
	APPENDIX.....	103
	REFERENCES.....	107

Introduction

1. Preliminaries

Low machine utilization, high proportion of waiting time from one step to another and high levels of expenditures in order to keep jobs flowing through the manufacturing facilities, are some of the major drawbacks of the traditional manufacturing systems. Due to this situation, it became obvious that something had to change and all the above problems had to be solved. At about the early 60's, it was pointed out that these requirements could be met with the aid of automated and computerized techniques (Buzacott, Yao, 1986). This observation led to the acceptance and implementation of a new technology, known as Flexible Manufacturing System (FMS).

Despite the fact that FMS pioneered in the United States, they were not readily implemented in the companies. American manufacturers failed to focus on the advantages of the new technology, as they did not realize all of the possible cost reductions and productivity improvements. Instead, they took the high risk and did not apply this system to their plants, even if they had to deal with a really competitive market. Many years had to pass, until 1986, when American companies following the influence of the Japanese, installed the first FMSs (Mohsen, 1992).

However, it is not the first time that automation is used in production. Factories used computer programs to guide the production. These programs did manage to control machines, which automatically performed applications as they were supposed to. Due to the technology evolution, automated guided vehicles were applied broadly, to large and complex in-plant systems. They are computer monitored, driverless vehicles which can move from one station to another in the plant without following a fixed path.

According to Mohsen (1992), many types of material-handling systems have been used in manufacturing, since 1920. Moreover, numerical control machines have been around since the 1950's. Buzacott and Yao (1986) though, pointed out that the first instance of the development of the basic concept of the flexible manufacturing systems is placed at about 1970.

In section 2, there is a definition of the flexible manufacturing systems and the reason why they are broadly used. In section 3, there is an extended reference of automatic guided vehicles which are a type of material handling devices which are commonly used in a flexible manufacturing system. Thereafter, in section 4, there is

a presentation of the ways with which an AGV system can be set up and its governing rules. Section 5 describes what exactly is modeling and simulation and why they are so useful. Next section (Section 6) includes the development of special modules with Extend software package. Finally, section 7 there are concluding remarks of both the literature review and the modules' development.

Hence, the aim of the thesis is to assist flexible manufacturing system simulations with software packages. This will be plausible by developing special modules into the simulation software, which will apply to the rules that conform to these vehicles.

2. Flexible Manufacturing System (FMS)

Generally, FMSs can be divided in two separate categories. The first one refers to the machine flexibility, whereas the second one refers to the routing flexibility. As far as the machine flexibility is concerned, it can either change the order of the procedures or it can change itself in order to produce new products. On the other hand, routing flexibility refers to the ability of alternating the programmed procedures depending on the situations.

The concept of an FMS is quite simple. It consists of many work stations, a transportation mechanism and a central computer facility. The work stations are interconnected by the transportation mechanism (material handling system-MHS) and all together, they perform under central computer control. In order this integration to be successful, information technology is employed. There is not a specific route that materials have to follow, and that is one of the major benefits that offers a flexible system (Buzacott, Shanthikumar, 1980). On the contrary, there are many alternative routes between each pair of machines that are selected by the computer, and one common material handling system which connects the machines with each other, and that is, after all, the reason why this system is called flexible.

However, there are other significant advantages too. FMSs enable systems create a wide product variety at prices that approach mass production. Machine set up times are getting shorter, the system is adjusted to changes fast, the inventories are being reduced and quality is improved. There are some other important advantages in production, too. Flexibility permits manufacturers to make a far broader product line. It, also, simplifies style changes and may stimulate a new product design. The production is not being held at large volumes but at competitive costs.

Despite these various benefits, there are some quite important disadvantages such as the number of machines that is fairly small (usually between 4 and 10 machines). The number of part types is also very limited for the planning period in consideration. But, the number of operations and tools can be so large that an implementing model of an FMS becomes unsolvable within a reasonable computation time (Sarin, Chen, 1987). Last but not least, high implementation costs and incompatibility problems between tools and operations have to be solved in order to cope with the integration challenge.

2.1. *A successful FMS implementation*

A successful implementation depends on the central core of an FMS which is the manufacturing information system that not only contains all the data, but also the procedural software. There are different parts of the system that work together, depend on each other, so they have to be properly integrated. Mohsen (1992) enumerates a series of basic steps for the appropriate implementation:

- 1.** Evaluate the “as-is” situation of the current IS, in order to develop a better “to-be” structure.
- 2.** Evaluate the company’s current situation in order to adopt new goals in terms of the competitive environment. However, this is a long-term investment so it must have the flexibility to be expanded in the future.
- 3.** Establish a desired outcome, set the aspects, and design the concept. It is essential to set both the desired outcome of the implementation and the requirements. The list of improvements, which are expected from automation, should include all the goals and the strategies of the implementation.
- 4.** Cost-benefit analysis; if the benefits do not exceed the cost of the implementation, then there is no point in implementing such an expensive system.
- 5.** Workers’ involvement; they have to be encouraged and explained about the reasons why they are under this change, in order to be convinced and become an active and effective part of the whole process.

Generally, long-run objectives have to be formulated, as this implementation is a long-term decision and these long-term plans should provide room for technological change and expansion. As it has mentioned before, the challenge of integration is

still vivid. Although the computers, robots, and sophisticated machine exist, the necessary software and common data bases that control and link the equipment are not yet sufficiently developed.

3. Automatic Guided Vehicles (AGVs)

Ben Saoud S. et al (2005) denoted that the major development in automation of the last decade resulted in the growing interest and use of FMS, which actually became really competitive against other manufacturing strategies.

The FMS competitiveness lies in the fact that it is plausible for many jobs to take place in the same FMS. Practically, this can be achieved with a better management and organization of the jobs, which take place not only in the system, but among the subsystems too.

Based on the previous, it is concluded that in order to exist the appropriate management and organization in the FMS, there have to be made several decisions. These decisions can be divided in two categories; pre-release and post-release. On one hand, pre-release decisions are related to decisions that are made before the process begins and include part types to be produced, buffers' type and size, pallets' number, fixtures' number and design. On the other hand, post-release decisions are related to decisions that are made after the process begins like alternative routing and scheduling material handling devices, such as the automated guided vehicles (AGV) (Ben Saoud S. et al, 2005).

AGVs can offer many advantages over push carts, forklifts trucks or fixed material handling devices. It is a fact that AGVs are more flexible than the non-computer controlled systems. However, the more automated become the systems, the more serious control problems arise. Egbelu and Tanchoco (1984) stressed that "The manner by which these problems are solved determines the operating effectiveness of the total system, as typified by an Automated Guided Vehicle (AGV)".

The first Automated Guided Vehicle application was developed for transporting groceries in a warehouse. While their initial applications have been in warehousing, there is growing use of automated guided vehicles (AGV) in the manufacturing area. AGVs are not only an important part of flexible manufacturing systems and warehousing, but they are also used for moving jobs as diverse as mail, laundry and hospital meals, public transportation, mining and airlines to cite only a few (Ganesharajaha T., Hall N. G. , Sriskandarajah C., 1998).

Sezen (2003) proposed two different ways of AGVs' use. The first one is when the AGV moves from station to station, being loaded with the item that has to be transferred to the stations. In this case, the vehicle is released only if the whole procedure for the item is completed. The second one is when the vehicle is used only as a transporter. It carries the item from station to station. Since the vehicle reaches its destination, it releases the item and returns to fulfill another task, while formerly, the AGVs had to serve exclusively the particular item until its process ended. As a consequence, in the first approach, it is required a great number of vehicles compared to the second one. What is more, General Motors Company is the pioneer of such an assembly system built in the U.S. with 185 unit-load carriers (Sezen, 2003).

AGVs are a type of handling system that plays an important role in the concept of an FMS (Ben Saoud, Jaoua, Bellamine-Ben Saoud, 2005). The reason why they are becoming more and more popular is because of the flexibility and adaptability that provide in an FMS environment. Benefits of building and using AGV systems include labor costs saving, flexible material handling, effective inventory control, greater quality assurance and safety, increased production, improved utilization of space, improved factory floor safety and flexible routing.

According to Hoff and Sarker (1998) an AGV is a driverless battery powered material handling vehicle. Thus, these self-propelled vehicles are automatically routed and monitored by a central computer (Ganesharajaha, Hall, Sriskandarajah, 1998).

In conclusion, it is obvious that AGV are not only hardware but software, as well. Qiu et al (2002) point out that the progress of hardware has far exceeded that of software since the 70's. However, Egbelu and Tanchoco (1984) stressed, that the progress of hardware was not large and that can be confirmed by the few technical publications on hardware developments. According to them, this happened due to the hardware's protection by corporate confidentiality and proprietorship.

3.1. AGVs yesterday and today

Japan was the leader in AGV system applications, with Europe to follow closely. In 1989, for example, Japanese companies purchased 5,000 vehicles, European companies 3,000 vehicles, and US companies only 500 vehicles. Today's AGVS is now in direct competition with other forms of industrial transportation (e.g. roller conveyors, belt conveyors, chain etc.). AGVS applications within the United States

have grown considerably and were expanding at a 30% annual rate during the '90's (Brazier, Shannon, 1987). Systems can be found in nearly every segment of industry. Where technically feasible, AGVS implementations are limited only by their operational considerations and economic viability.

The Volvo industry in Kalmar, Sweden was the first one in history that integrated a large-scale manufacturing application of an AGV system in its production in 1974, whereas General Motors (GM) made the largest application of AGV in North America within a decade after Volvo. In addition, there is a large AGV application at GM's Doraville, Georgia plant, where there are 135 AGVs which carry workpieces through dozens. What is more, in Rochester, New York, where Kodak's main distribution centre is placed, there are employed 60 AGVs in the plant's procedures. Finally, New York Times employs 23 AGVs transporters, which facilitates printing 45,000 newspapers per hour by 6-color presses (Ganesharajaha, 1998).

The widespread use of AGVs can be verified by a survey, conducted in July, 2009. The survey's conductors (RMT Robotics, LTD and Modern Materials Handling magazine) turned to the materials handling professionals, by setting the question how they prefer to carry items or move them from station to station in a plant. The results were quite expected, since 59% of them acknowledged the major role of the AGVs in manufacturing, warehouse and distribution operations.

Since their implementation, much attention devoted to the technology and rapid progress has been witnessed. Today, the technology of AGVs has moved on. Despite being monitored by a central computer, they have a powerful on-board minicomputer. They are being guided locally and as a result they are becoming more and more flexible.

Although the AGVs used to travel on a physical guide path, the evolution of technology led to the virtual flow path. This path is much more flexible than the former one, as it is easy to perform changes, without significant difficulties (Ujvari, Hilmola, 2006).

3.2. *Advantages*

Ganesharajaha (1997) supports the opinion that AGV systems have become an important material handling method. The primary advantages occur in flexibility, space utilization, safety and overall operating cost.

Flexibility: There is a specific fixed route that it should follow, but if anything happens it is able to change this route and go from another way. They work on a

path (guide path or flowpath) that does not interfere with the others. They are so flexible that, if anything happens their guide path can be modified in a matter of minutes. Even if they are wire-guided vehicles can be dynamically rerouted to respond to changing priorities within an existing system.

Space utilization: Unlike conveyor belts, AGV occupy workspace only when they are operating in a given area or when they are parked. After they fulfill their mission, they move to another workstation. The AGVs are reassigned to another mission if there is any unattended handling task in the facility (Egbelu, Tanchoco, 1984). Otherwise, if there is no unattended handling task, the vehicle is set idle until a new task appears. On the contrary, conveyor belts are placed in a specific space. Ganesharajaha et al (1998) also highlight that since there are no physical barriers with AGVs (unlike the conveyors), the Automatic Guided Vehicle Systems can share their space with other uses such as pedestrian or forklift aisles, resulting in the improvement of the factory's space utilization.

Safety: AGVs are much safer than other competitive technologies, which were broadly used in the past. This refers to the lights and horns which the AGVs are equipped with. This makes them visible to the pedestrians and makes an accident less possible to occur. What is more, there is no possibility of an operator error, unlike forklifts. Even if though, a single-point failure occurs (eg. a vehicle fails) system can continue operate without problem.

In a large number of today's modern manufacturing and transport systems, AGV systems are the material handling system of choice because of the general belief that they include increased flexibility, better space utilization, improved factory floor safety, easier interface with other automated systems, increased productivity, product quality, system flexibility and thus lower costs.

D. Butler of Caterpillar Corporation has quoted as follows: "The opportunities for cost savings when reducing inventories do not stop with reduced carrying costs and lower capital investment. Savings in reduced floor space, insurance costs, indirect labor, storage equipment, and many other areas also can be realized."

3.3. *Disadvantages*

As it was mentioned before, AGVs are so flexible that if anything happens their guide path can be modified in a matter of minutes and can be dynamically rerouted. However, there is a problem that is possible to occur. It is possible in the system that at a specific time there are many demand points for the same vehicle. This is a

serious operational control problem that has to do with a decision; which task the vehicle will finally serve. This situation can also be met vice versa. Thus, a task arises while the system has many vehicles available and ready to serve. Now, the operational control problem has to cope with another decision; which will be the vehicle that will be assigned for the task. Vehicle distribution in the network is an operational control problem (Egbelu, Tanchoco, 1984).

Every decision will, finally, affect material flow, machine utilization, buffer storage requirements and vehicle effectiveness. Consequently, these interrelated decisions and the variety of the alternate choices in the system, make the problem quite complex. This complexity can definitely affect system's cost and quality.

As far as cost is concerned, although AGV systems pay for themselves rapidly by reducing unit operating costs, they are significantly more expensive than other material handling technologies.

AGVs in the past were few and as a result their routing and scheduling was trivial, unlike today (Qiu et al, 2002). It is shown that inadequate software of an AGV system can cause congestion or deadlocks in AGV operations. Today's technology development demands large fleet of AGVs (even hundreds of them), that work simultaneously and are involved in multiple tasks. Another relative problem that has been observed is that the even if the computing capacity of PCs is considered to be high, AGV simulation still takes long execution time (Ujvari, Hilmola, 2006). However, in many cases that the production system is crowded, there is no need for speed, since production labor also needs space, even if speed is argued to be the important factor in future production system improvement plans.

Ross et al (1996) stressed that AGVs control has become difficult and costs a lot, due to the evolution of technology. This happens because, when a system has to be upgraded, two separate problems arise. The first one has to do with the upgrade itself, which sometimes makes the system less effective, due to its implementation and controlling complexities. There are systems though that cannot be easily modified with a new technology and even if this is plausible, it is very expensive. So, the challenge is to design AGV systems that maintain a given performance level while simplifying control and modification.

4. Setting up an AGV System

The role of the AGVs in the material handling systems has proved dominant and very important both as regards the systems themselves and the FMS. This is the

reason why special attention has been given to the control and design of a system which contains AGVs.

Thus, before setting an AGV system, it is essential to decide certain issues concerning both design and control. During the design, it must be decided the possible vehicle routes and the number of vehicles that can accommodate (guidepath layout and fleet size). As far as control concerns, the issues arising are related to the vehicles' dispatching, routing and scheduling (Ben Saud, 2005).

According to Reveliotis (2000) the main reason for the systems' design and control problems is the continuous development of the technology and the new capabilities that can offer. Thereby, design and control are problems that are adapted to the specific technological and operational requirements.

4.1. *Design of an AGV system*

Duinkerken et al (2006) complemented the words of Ben Saud (2005) and stressed that the design of an AGV system includes not only the system's equipment (i.e. vehicles) but also its operational control. Consequently, during the system's design, decisions on vehicle dispatching, scheduling and routing are the heart of the operational procedure. In current systems, there is a map that provides all the available routes for every vehicle. These routes are fixed, yet they can be combined with each other, in order to obtain a complete guide path for them so as to be capable to move from their initial to the demand point (Reveliotis, 2000).

The guide path though is formed according to the locations of the pickup and delivery points. In fact, it is very important the allocation of these points, because that will determine the number of the available routes, their length and their transfer intensity. Actually, these design issues depend on the type of the systems that needs to be designed. Hence, the positions of the pickup and delivery points can either be dynamic or fixed.

4.2. *Control of an AGV system*

As it is already mentioned, a flexible manufacturing system consists of a set of workstations performing a number of different operations, integrated under a common automated transportation system. All the operations in these systems happen at the same time. As a consequence, if any changes occur, the system reacts fast and is adjusted in the new circumstances, in contrast to the traditional methods

which are inflexible regarding changes. Careful attention, though, must be paid to the dispatching, scheduling and routing in a system, as the cost of using the resources of an FMS is quite high (Kimemia, 1983). In addition, these systems are often quite large and complex, and consequently there are some issues arising during the manufacturing process that will be analyzed in the following.

Duinkerken et al (2006) define dispatching as “the process of assigning a transportation job to an AGV”, where scheduling is “the process of dispatching a set of AGVs to a batch of transportation jobs”. Routing though, is “the process of determining routes for a set of AGVs to fulfill their respective transportation jobs”.

4.3. Vehicle dispatching

The vehicle dispatching decisions can be divided in two separate categories. The first one assumes that when a task is generated in the facility, there is more than one available vehicle which can serve. Thus, the decision is to determine which vehicle from the idle ones will eventually serve, when there is only one demand point.

The second decision is the exact opposite from the former one. Hence, it is assumed that there are a specific number of work centers and one vehicle (or one idle vehicle) in the facility. Thus, the decision is to determine which work center the available vehicle will serve, while all these work centers request a vehicle simultaneously.

In both cases, the decision is to set priorities either to the vehicles or to the workstations and dispatch the vehicle or serve the workstation with the highest one (Egbelu, Tanchoco, 1984).

4.4. Vehicle Scheduling

Graves (1981) defines vehicle dispatching as the “allocation of available production resources over time to best satisfy some set of criteria”. Dantzig and Ramser (1959) though, were the first who formulated the vehicle dispatching problem. In fact, with the use of linear programming, they developed a heuristic approach. Graves (1981) categorized scheduling according to three aspects:

- a. Requirements generation (open shop versus closed shop)
- b. Processing complexity

c. Scheduling criteria

a. Requirements generation (open shop versus closed shop)

This aspect refers to the requirements that a customer may have, or to the decision for replenishing the inventory. Alternatively, it refers to the decision between an open shop and a closed shop. In an open shop, there is no stock, as it is produced the exact amount that a customer requests. On the contrary, in a closed shop, the production is not adjusted by the customers' orders. Their requirements though, are served by the stock, and production is adjusted according to the replenishing decisions of the stock. Scheduling is involved in both cases.

Besides this differentiation, requirements generation can either be deterministic or stochastic depending on how they are actually generated, no matter whether the shop is closed or open. For instance, the processing time for each step of the task may be a random variable with specified probability distribution (stochastic), or may be known (deterministic).

It is respectively the same problem between a static and dynamic environment. In a static environment all the system requirements are fixed and clearly defined. That means that these requirements cannot be changed by being removed or added. Alternatively, a dynamic environment is more flexible. It has no fixed requirements and can be extended in the future.

Most scheduling systems though, are implemented in a deterministic and static production environment for a specific time period.

b. Processing complexity

This aspect refers to the number of processing steps associated with each production task. This in turn, can be classified in the following categories:

i) One stage, one processor (facility)

This is the simplest and most popular problem form. It is also known as the one-machine problem. There is only one task that has to be done and one machine that completes the process.

ii) One-stage, parallel processors (facilities)

This is a generalization of the one-machine problem, where the only existing task has to be done by any of a number of available machines. It has received less attention than the single facility. The one stage with parallel processors problem has more theoretical than practical interest, even if it is closer to the reality than the previous one.

iii) Multistage, flow shop

Flow Shop Scheduling Problems, are a class of Group Shop Scheduling Problems in which the operations of every job must be processed on machines in a same order. This is the simple multistage scheduling problem. In this case, there are many tasks and each task requires processing at different work centers. In addition, this problem assumes that there is a strict priority over which step will follow the other in the procedure. Flow Shop Scheduling Problem is used in many facilities, one of which is computer installations.

iv) Multistage, job shop

The job-shop problem is a discrete problem and a generalization of the famous travelling salesman problem. It is a prominent illustration of a class of problems in computational complexity theory which are hard to solve. This is the most general production scheduling problem among the former ones in the classification. The processing steps need to have neither priorities nor specific routes that have to follow for a task. The problem falls into two decisions. The first one is to schedule properly the production so as to satisfy the given requirements, whereas the second one is based on the decision of scheduling the production at a minimum cost. This category may be closer to the reality due to its complexity.

c. Scheduling criteria

The third aspect includes the evaluation of the scheduling decisions. There are two separate criteria used for this evaluation; schedule cost and schedule performance criteria. The first one occurs once the schedule is generated and the process has to be monitored. Furthermore, Graves (1981) denotes that this category also includes fixed costs, inventory holding costs, shortage costs for stocking out or expired deadlines and costs for a possible schedule implementation in a dynamic environment.

The schedule performance class includes the utilization level of the machines and work stations, the number of tasks that were late, the time period that some tasks were late, and the flow time of the tasks.

4.4.1. Scheduling rules

A typical machining center in an AGV system consists of one or more machines, a queue for the items that are inserting in the machining center and a queue for the items that are exiting the center, since the procedure is completed. All rules assume

that the machines are identical and the queues have enough capacity to accommodate the incoming or outgoing items.

Egbelu and Tanchoco (1984) stressed some scheduling rules according to which vehicles can be scheduled. These are heuristic rules that are employed and assign priorities to the vehicles and include the following.

- Random Vehicle (RV) rule

Each vehicle is randomly chosen to complete a task. This selection is made without taking into account any specific characteristic of the AGVs.

- Nearest Vehicle (NV) rule

The vehicle which is chosen to complete a task is the one, which happens to be nearest to the process that needs to be completed. Actually, this rule can be viewed by two different aspects. The first one is in terms of distance, whereas the second one is in terms of time. The vehicle which is finally scheduled to be sent is either the one which is the closest to the final destination, or the one to which it takes the least to reach this destination. However, in a congested network, distance and time travel rules are not equivalent.

This is equivalent, though, to dispatching idle vehicle j such that:

$$d_j = \min \{ d_i \} \forall i, \text{ where } i \text{ is the idle vehicle and,}$$

d_i is the distance between the idle vehicle and the demand point.

Alternatively, if s_i is the travelling speed of the vehicle, then let

$$d_j/s_j = \min \{ d_i/s_i \} \forall i, \text{ where } i \text{ is the idle vehicle}$$

This expression proves that for a congested traffic network, the shortest travel distance path is not necessarily the shortest travel time rule. For example, suppose that $d_j < d_i$. If $s_j > s_i$ then $d_j/s_j > d_i/s_i$.

- Farthest Vehicle (FV) rule

This is exactly the opposite rule to the above. The system picks the vehicle which is the least near to its demand point. This schedule option has not a directly benefit to the system but according to Egbelu and Tanchoco (1984), "...it does give the system designer what effect unnecessary vehicle travelling distance could have on

system handling effectiveness". In this case, in a congested network, distance and time travel rules are equivalent.

Thus, we get

$$d_j = \max \{ d_i \} \forall i, \text{ where } i \text{ is the idle vehicle and,}$$

d_i is the distance between the idle vehicle and the demand point.

Alternatively, if s_i is the travelling speed of the vehicle, then

$$d_i/s_j = \max \{ d_i/s_i \} \forall i, \text{ where } i \text{ is the idle vehicle}$$

- Longest Idle Vehicle (Longest IV) rule

This classification denotes a picking choice depending on how much time a vehicle has been idle. More specifically, it depends on the longest time that a vehicle has been idle. This offers a balance in the system, as it gives the opportunity for all vehicles to participate in the process.

In this case,

$$t_j = \max \{ t_i \} \forall i, \text{ where } i \text{ is the idle vehicle}$$

$$\text{and } t_i = t_b - t_a,$$

with t_b = the present time or the time that the vehicle is decided to start and,
 t_a = the time that the vehicle was set idle.

Of course, the time that the vehicle is decided to start (or the present time) must exceed the time that the vehicle was set idle in order to obtain a logical result.

- Longest Cumulative Idle Vehicle (Longest CIV) rule

In spite of dispatching the vehicle considering its idle time between two uses, it is quite practical to calculate the entire time that the vehicle had been remained idle or until a job occurred.

Hence,

$$t_j = \max \{ \sum t_i \} \forall i, \text{ where } i \text{ is the idle vehicle}$$

$$\text{and } \sum t_i = \sum (t_b - t_a),$$

with t_b = the present time or the time that the vehicle is decided to start,
 t_a = the time that the vehicle was set idle and
 $\sum (t_b - t_a)$ = the total time that the vehicle remained idle.

The time that the vehicle is decided to start (or the present time) must exceed the time that the vehicle was set idle in order to obtain a logical result.

- Least Idle Vehicle (Least IV) rule

This classification denotes a picking choice depending on how much time a vehicle has been idle. More specifically, it depends on the least time that a vehicle has been idle.

In this case,

$$t_j = \min \{t_i\} \forall i, \text{ where } i \text{ is the idle vehicle} \\ \text{and } t_i = t_b - t_a,$$

with t_b = the present time or the time that the vehicle is decided to start and,
 t_a = the time that the vehicle was set idle.

Once again, the time that the vehicle is decided to start (or the present time) must exceed the time that the vehicle was set idle in order to obtain a logical result.

- Least Idle Cumulative Vehicle (Least CIV) rule

Likewise, instead of dispatching the vehicle considering its idle time between two uses, it is quite useful to calculate the entire time that the vehicle had been remained idle during the simulation run or until a job occurred.

Hence,

$$t_j = \min \{\Sigma t_i\} \forall i, \text{ where } i \text{ is the idle vehicle} \\ \text{and } \Sigma t_i = \Sigma(t_b - t_a),$$

with t_b = the present time or the time that the vehicle is decided to start,
 t_a = the time that the vehicle was set idle and
 $\Sigma(t_b - t_a)$ = the total time that the vehicle remained idle.

It is essential to note, though, that the time that the vehicle is decided to start (or the present time) must exceed the time that the vehicle was set idle in order to obtain a logical result.

- First-In-First-Out Vehicle (FIFO) rule

The FIFO rule requires that the entering and exiting of the queue will be performed in a FIFO order, i.e. if the vehicle i is assigned for a job before vehicle j , then vehicle i must be dispatched before vehicle j .

- Last-In-First-Out Vehicle (LIFO) rule

The LIFO rule requires that the entering and exiting of the queue will be performed in a LIFO order, i.e. if the vehicle i is assigned for a job before vehicle j , then vehicle i must be dispatched after vehicle j . The order of the vehicles that pass in and out the queue reminds of a stack. That is, the first vehicle that passes in, is the last that will be dispatched.

- Least Utilized Vehicle (Least UV)

One of the main metrics that are available in most computer controlled AGV systems is the utilization of each vehicle. This rule benefits from this fact and chooses to schedule the dispatching using the vehicle that has the lowest utilization. This rule is equivalent with the Longest Idle Vehicle rule.

In this case we have,

$$U_j = \min \{U_i\} \forall i, \text{ where } i \text{ is the idle vehicle,} \\ \text{and } U_i \text{ is the utilization of the vehicle } i$$

4.4.2. Some general comments

Despite significant progress in production scheduling methods, there are still some issues that mainly concern how to improve those. What is more, real environments constantly change, making it difficult or even impossible to be classified according to their scheduling method. Apart from the environments' volatility problem, it was pointed out by Graves (1981) that "practitioners have less incentive than theoreticians to report their work in production scheduling and this is due both to the perceived limited generality, and to the proprietary nature of their work".

Graves (1981) also stated that the main problem is not the production scheduling, but the production rescheduling. Hence, as soon as the system is scheduled and operates, all its variables will change. Consequently, the system cannot operate anymore, as it will attempt to use the initial data. Thus, it is necessary to implement a schedule without demanding certain and rigid environment conditions. On the contrary, the more flexible and adaptable to the environment's changes is the schedule, the more suitable becomes.

4.5. Vehicle Routing

The vehicle routing problem involves the existence of a number of vehicles, a number of customers with specific requirements and a set of possible routes for the vehicles that start and end in a central warehouse. Also, this problem faces the following constraints; each customer will be served only once and his requirements cannot exceed vehicle's capacities (M. Solomon, 1987).

First of all, vehicle routing defines the destination. The procedure must give the answer to the question, which work centre should visit the first available AGV. The second part of the routing is to define the best path, of the available ones, in order to reach the demand point without conflicts.

Reveliotis (2000) denotes that in the past, conflictions were avoided by implementing non-overlapping subsystems of routes in the main system, each of which was served by one vehicle. In recent years though, conflictions can be avoided with different ways. So, there are either sensors that detect if there is any obstacle in front of the vehicle or control zones. Practically, as far as the last one is concerned, these control zones are a network of guide path. This network is divided in several network subsystems which actually are logical zones. In this case, the confliction is avoided in two ways. On one hand every logical zone can be served by only one vehicle, whereas every vehicle has sensor that warns it of the possible existence of another vehicle. If there is another vehicle, the sensor makes the vehicle to stop.

As far as the types of vehicle routing algorithms, Qiu et al (2002) suggested the following classification:

- a. Algorithms for general path topology
 - i. Static method
 - ii. Time-window-based methods
 - iii. Dynamic methods
- b. Algorithms for path optimization
 - i. 0-1 Integer-programming model
 - ii. Intersection graph method
 - iii. Integer linear programming model
- c. Algorithms for specific path topologies
 - i. Linear topology
 - ii. Loop topology
 - iii. Mesh topology
- d. Algorithms for dedicated scheduling algorithms

a. Algorithms for general path topology

This type of algorithm is quite simple, as it tries to find possible routes for the AGVs, without regarding the general path topology. This is an important problem, as there are several differences between vehicle routing problem and AGV routing problem. In the first case, as Qiu et al (2002) denote, the moving vehicles “are regarded as moving points”, making it impossible for them to collide to each other. In the second case though, where vehicles are not considered as moving points, the paths cannot guarantee that AGVs can coexist and run next to each other.

Therefore, there have been adopted several methods that can be divided as follows:

i. Static method

This method is usually used when the problem is quite simple and there is no reason for any complexity. Although the method benefits from simplicity, it suffers from the lack of optimal solutions. The latter are not guaranteed.

Static method demands the entire path to remain occupied until a vehicle completes its tour. Consequently, the path cannot be useable by new added AGVs until the older ones reach their destination. In other words, the algorithm does not allow vehicles to use path resource that could otherwise be shared during different time-windows. So, even if a path exists, the procedure cannot always find it. This is the reason why this method is suitable for small networks with a small number of vehicles.

ii. Time-window-based method

This is an advanced method when compared to the previous one. The basic benefit of this method is that it uses more efficiently the paths enhancing their utilization. This is possible by segmenting the paths, so as many vehicles use the same path, yet in different time-windows. Therefore, time-window-based method is suitable for AGV systems.

iii. Dynamic method

Dynamic method offers a dynamic determination of the utilization of any path. It is a very flexible method when compared to the former ones. Static and time-window-based method are cannot route the AGV quickly than the dynamic one can. This is due to the fact that, the two first methods decide for the routing even before it happens. So, before the actual routing decisions are made related to which vehicle

will be sent to the demand point and which path will be an optimal solution for the routing.

Thus, Taghaboni and Tanchoco (1995) proposed dynamic routing techniques, so as to speed up the whole process. They also, proposed a dynamic routing technique, namely incremental route planning, "...which can route AGVs relatively quickly compared with some static algorithms". The AGV must travel from its initial position to the demand point, passing through the existing nodes in the network.

This routing technique uses information about both the status of neighboring nodes and the global network, and respectively decides which node the vehicle will pass through next. The whole procedure is repeated until the vehicle reaches its destination. The algorithm can work in networks with one and two directions.

However, in case that there is no longer a simple network, as it was assumed, the planning route is not longer high efficient, too. Subsequently, the algorithm may predict neither the most efficient route nor the delays.

b. Path optimization

As the calculation of the optimal route in a network is quite time consuming, many researchers propose the idea of the optimization of path layout or the distribution of the stations. The optimization problem is usually formulated in integer programming models as discussed in Qiu et al (2002).

i. 0-1 Integer-programming model

The objective of this modeling approach is to find an optimal path network which will minimize the vehicles' total travelling distance. However, the computation of the optimal network has several drawbacks. First of all, every calculation is made for the loaded vehicles' total travelling distance. There are though, unloaded ones, as well. This problem, may affect the routing control and system throughput. In addition, this model considers AGVs to run along the same route and have with the same origin and destination every time. As a result, when AGVs with different origins and destinations run simultaneously, they may conflict with each other.

Another problem is that with this model it is more possible to be obtained a non empty solution for the optimal routes, rather than the opposite. What is more, the number of 0-1 variables is significantly large, making the computations a major problem.

However, if the stations' locations and the path designs are given, then a new enhanced method can be implemented. This method, intersection graph method, is

able to compute the best path design among the other paths in the network. Nevertheless, the new method has focused on the computations speed rather than the quality of results (path design). So, there are still problems that have to be outreached.

ii. Intersection graph method

The advantage of this method is that it uses only a sub network of the entire path network to find the best solutions. In fact, it takes into account only the intersection nodes for its calculations. Consequently, the computation's amount is quite lower than it was at the other methods. This fact enables intersection graph method to be used in large AGV systems. Yet, these reductions in calculations may not obtain all the existing optimal solutions.

iii. Integer linear programming model

Integer linear programming algorithm is used in this method to obtain the best solutions for both the path layout and the vehicles' allocations. In contrast to the 0-1 Integer-programming model which did not considering to its calculations the unloaded AGVs, this model does calculate both (loaded and unloaded). As a consequence, this method becomes quite suitable for large AGV systems.

c. Algorithms for specific path topologies

Algorithms for these specific path topologies usually are more effective than those for general path topology.

i. Linear topology

Linear topology is a topology consisting of a main path with a terminator at each end. All nodes (workstations) are connected to the linear path. This path topology is quite simple and thus, it is considered to be basic type of path layouts. Because of its simplicity, it can be implemented efficiently in any system, whether is small or not. The conflicts are rare and the procedure can be completed within a very short time.

ii. Loop topology

Single-loops, multi-loops and segment floor topology are only a few examples of the general loop topology. In the case of a single-loop topology, there is only one loop, which allows several vehicles to run into. This topology is quite simple and if it is assumed that all the running vehicles have the same speed and direction, then no

collisions will take place. However, Qiu et al (2002) argue that single loop is not a suitable topology in case of a quite large system, which actually is closer to reality.

To confront with this problem, it can be placed to the loop only one vehicle. Hence there is no possibility for conflicts, as there is only one vehicle which runs in the loop. In this case though, the system's throughput will be remarkably low.

Alternatively a multi-loop topology could be a solution. Still, there is a problem; the large moving costs, when a vehicle moves from a loop to another, make it forbidden for the system to be large.

At last, segment floor topology is suitable for three types of network; connected, partitioned and split-flow. Every segment is served by one vehicle, which can move in both directions. Also, there are two devices placed at the ends of every segment. These devices assist the vehicles to move from segment to segment. Despite the fact that this topology is considered quite simple, it faces high costs when the devices transfer the vehicle through segments. What is more, this transfer is often time-consuming and causes delays in the system.

iii. Mesh topology

This topology implies that every node of the network is connected with all the remaining nodes. In spite of the network's complexity, the algorithm is not affected by the mesh's size. However if the number of vehicles is not the suitable for the size of the mesh (they are located at a distance from one another), the algorithm may not obtain a best solution.

d. Dedicated scheduling algorithm

Akturk and Yilmaz (1996) proposed a mixed-integer programming algorithm. This algorithm takes into consideration, at the same time, both vehicles and jobs. Hence, if there are many vehicles and jobs, the computations for an optimal solution may be quite complex. As a consequence, it is not suitable for large systems.

It is true that today's automation and computerization have solved many problems on vehicle automation. However, there is no one who can guarantee that, despite the automation, there will be no congestions and deadlocks. Thus, the problem of scheduling and routing remains.

5. Modeling and Simulation of AGV Systems

A model is a logical description of how a system performs. Simulation is the design of a system's model and the experiments that are carried out on it. This is the reason why simulation is also considered as the act of imitation. Models are used to describe how a real-world activity performs. They also, enable somebody to test hypotheses at a fraction of the cost of actually undertaking the activities which models simulate (Diamond et al. 2002).

AGV systems often use computer simulation as a tool for their analysis. AGVs interact with many parts of the systems or even with themselves. Thus, simulation is the appropriate way for the presentation of these interactions (Davis, 1986).

The time of a simulation is not often the real time, but an abstraction of it. So, the time passes and changes occur in the system. If the simulation model is valid, the simulation will actually reflect the behavior of the real system.

Many AGV design problems are solved with mathematics programming procedures, many of which are based on the transportation problem. Nevertheless, sometimes, the size of the applications is so large that they require long computer runs. Thus, simulation, which can be a detailed description of the real system, is used as a powerful tool for the analysis of AGV systems.

In general, there are many strategies and techniques which were used by a number of researchers for scheduling and dispatching (Naish, Croft, Benhabib, 2003). As it is already mentioned, AGV dispatching rules can be classified into two categories: work-centre initiated and vehicle-initiated rules (Egbelu and Tanchoco 1984). Work-center initiated rules are related with a certain decision; which AGV will be assigned to serve the work center among the idle ones (Shneor et al, 2006).

This thesis is concerned with several general scenarios which are based on how to choose the appropriate vehicle. These are:

1. Least Idle Vehicle (Least IV) - the vehicle which has remained idle the least since its last job is dispatched. Essentially, this algorithm works with the aid of departure and arrival time of the vehicle. Departure and arrival time have reference point the initial block that AGV was in and waited a move request. If these times are recorded, it is quite simple to calculate the idle time. The vehicle with the least idle time is chosen and dispatched.
2. Least Cumulative Idle Vehicle (Least CIV) - the vehicle which has remained idle the least since the shop opened. This algorithm calculates all the times that vehicles have been idle, during the operation of the shop. Practically, it is based on the idea of dispatching the AGV which was totally idle the least.

3. Longest Idle Vehicle (Longest IV) - the antithesis to the Least IV rule. The procedure here is quite similar to the Least IV procedure. The difference is the fact that the vehicle with the longest idle time is chosen and dispatched.
4. Longest Cumulative Idle Vehicle (Longest CIV) - the antithesis to the Least CIV rule. The algorithm calculates all the times that vehicles have been idle, during the operation of the shop. It is based on the idea of dispatching the AGV which was totally idle the longest.
5. Least Utilized Vehicle (LUV) – the vehicle which was used the least is dispatched. Therefore, this rule ensures uniform work load on the vehicles circulating in the system (Mahadevan, Narendran, 1994). Egbelu and Tanchoco (1984) pointed out that this rule takes advantage of the fact that in most computer controlled AGV systems, time persistent statistics on the utilization of each vehicle are maintained.
6. Nearest Vehicle (NV) – in a system using NV, a load enters the systems and places a move request. Thereafter, the shorter distance along the travelling paths to every available vehicle is calculated (De Koster et al, 2004). Consequently, the vehicle with the shortest travelling distance is dispatched.
7. Farthest Vehicle (FV) – this is an antithetical to the NV rule. As soon as the move request is placed by a load, the farthest distance along the travelling paths to every available vehicle is calculated.
8. First In First Out Vehicle (FIFO) - according to this rule there is a queue priority. The vehicle that comes first is dispatched first too.
9. Last In First Out Vehicle (LIFO) - the vehicle that comes last is the one which is prioritized to be dispatched. This rule operates like a stack. The last piece that will be placed on the stack will leave first.
10. Random Vehicle (RV) - the pick-up task is randomly assigned to any available vehicle in the shop. There is no specific rule that is followed. This is done without regard to the relative location of the vehicles to the point where the vehicle is needed (Egbelu, Tanchoco, 1984).

Practically, the goal of this thesis is to deal with the problem of controlling a vehicle and particularly the scheduling problem. In order to confront with this challenge, it was essential to simulate the process of production in an industry with a message-based discrete event simulation software.

Subsequently, all of the above concepts have been modeled in the Extend v6 simulation software package. Particularly, it was created a number of hierarchical blocks for every different concept. A hierarchical block is a block that contains several blocks from the default libraries of Extend software or even another hierarchical

block, too. There is a group of blocks nested inside. These blocks represent a portion of the model, or a subsystem. At the end, all these hierarchical blocks were inserted in a library, in which anyone can refer to, in case it is a need use them in a simulation.

6. The Models

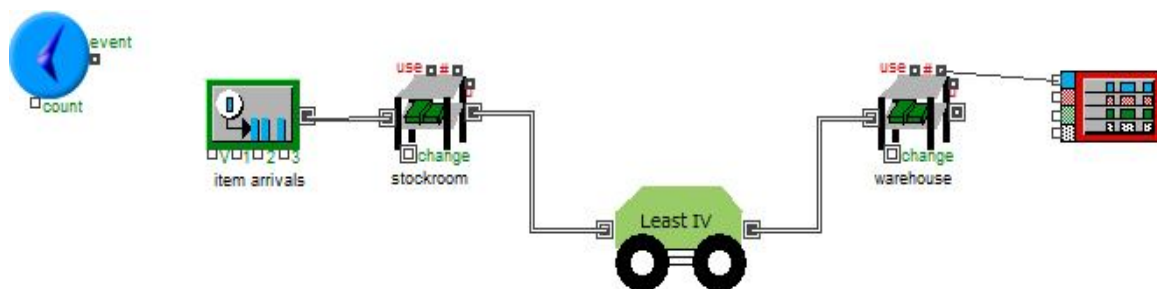
All cases, which will be described, have the same scenario. There is an industry which produces yellow cups. As far as the production of the cup is completed and before being offered for sale, there is one thing left; each cup must have the firm label on its bottom. Since the label is stuck on the bottom, the cup is transferred to a warehouse, ready for a sale. The whole process is presented in detail, below.

All the ready cups are placed on a conveyor belt ("item arrivals" block). This belt leads the cups to a route which is connected with the AGVs. At this point, cups wait at the stockroom for the appropriate AGV ("stockroom" block). This AGV is chosen depending on each algorithm that is used. AGV is assigned to do a specific job; it has to receive the unlabeled caps and deliver them to a machine that sticks the firm label to their bottom. At this point, there is a vehicle scheduler whose function is to select one vehicle to perform the route and complete its assigned job (hierarchical block). After the work is done, AGV receive the labeled caps and transfer them to the conveyor belt. Finally, after they are released, cups are transferred ready for sale to the warehouse ("warehouse" block).

Below, there are the rules, according to which the vehicle scheduler operates and the appropriate vehicle is chosen.

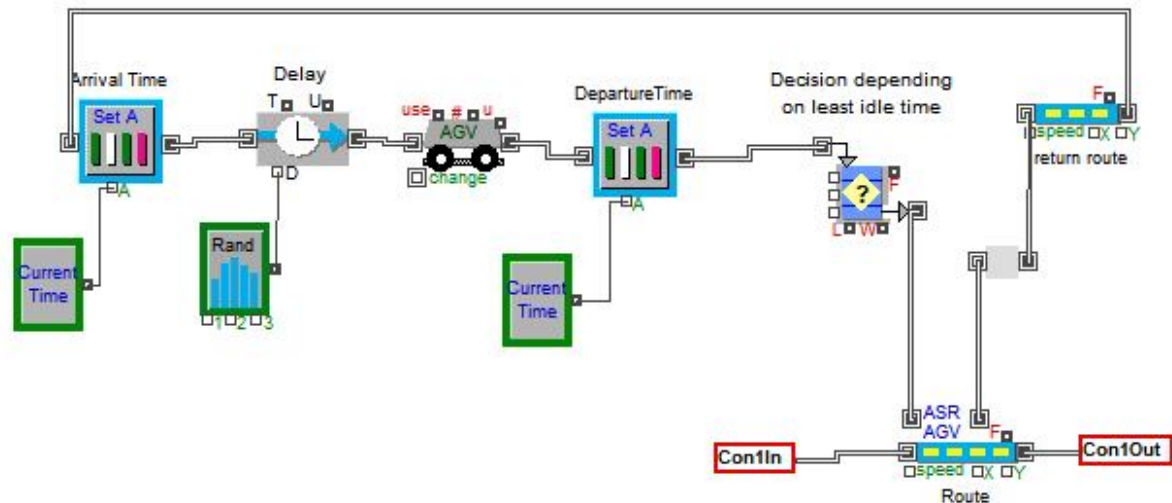
a. Least Idle Vehicle (Least IV)

i. About the Least IV model



Picture 1a – **The Least IV model**

The objective of the model is to calculate the idle time of the vehicles and dispatch the one with the least idle time. As it has already been presented formerly, by the time the cups arrive at the hierarchical block "Least IV", a separate process begins. Picture 1a shows the entire model, whereas picture 2a shows the subsystem of the model inside of the hierarchical block "Least IV".



Picture 2a – The "Least IV" hierarchical block

The "Con1In" icon represents the input connector of the "Least IV" block. Respectively, the "Con1Out" icon is the output connector of the block. Since the cups have started their route from generator block ("item arrivals" block), they pass through the "Con1In" connector and wait for an AGV.

AGVs begin their travel from the "AGV" block. The first AGV, which will travel, is the one that will also serve, as all vehicles are idle and there is no reason to decide which one has been idle most. As soon as it arrives to the conveyor belt ("Route" block), receives the unlabeled cups, completes its job and releases the labeled cups. The labeling process has no particular importance and this is the reason why it is defined to take place inside the "Route" block.

Additionally, the vehicle goes back to where it started through a different route ("Return Route" block). The grey blank block between "Route" and "Return Route" block does not affect the system, yet it is used for the animation's maintenance. During the return it passes through a "Set Attribute" block ("Arrival Time" block) which sets attributes to the items that pass through.

Practically, this block is connected with a system variable block ("Current Time" block) and records the exact time that AGV finishes its travel (Arrival Time). After that, it enters the "AGV" block and remains idle until a new job arises.

If there are arrivals of cups, AGVs begin to travel to receive them. The time that they begin their route, they pass through the "Set Attribute" block and another time stamp is being recorded. This time stamp is the Arrival Time of each vehicle.

As a matter of fact, the "Set Attribute" block that records the arrival time is exactly at the entrance of the AGV block, whereas the "Set Attribute" block that records the departure time is exactly at the exit of this block. The "Delay" block symbolizes a random delay that the vehicle can possibly encounter, during the way in and out process from the block.

All these vehicles, that begun their route due to the item arrival, are gathering in a block named "Queue Decision" block. "Queue Decision" block is the heart of the algorithm.

The Least Idle Vehicle algorithm assigns the highest dispatching priority to the vehicle which has the least idle period among all available ones. The dispatching process can be expressed by selecting a vehicle, such that the abstraction of the Arrival Time from the Departure Time gives the lowest result.

"Queue decision" block releases items according to the results of the equation defined in it. There are two different inputs that were set as parameters in this equation, Arrival and Departure Time. Hence, the result is calculated from the equation:

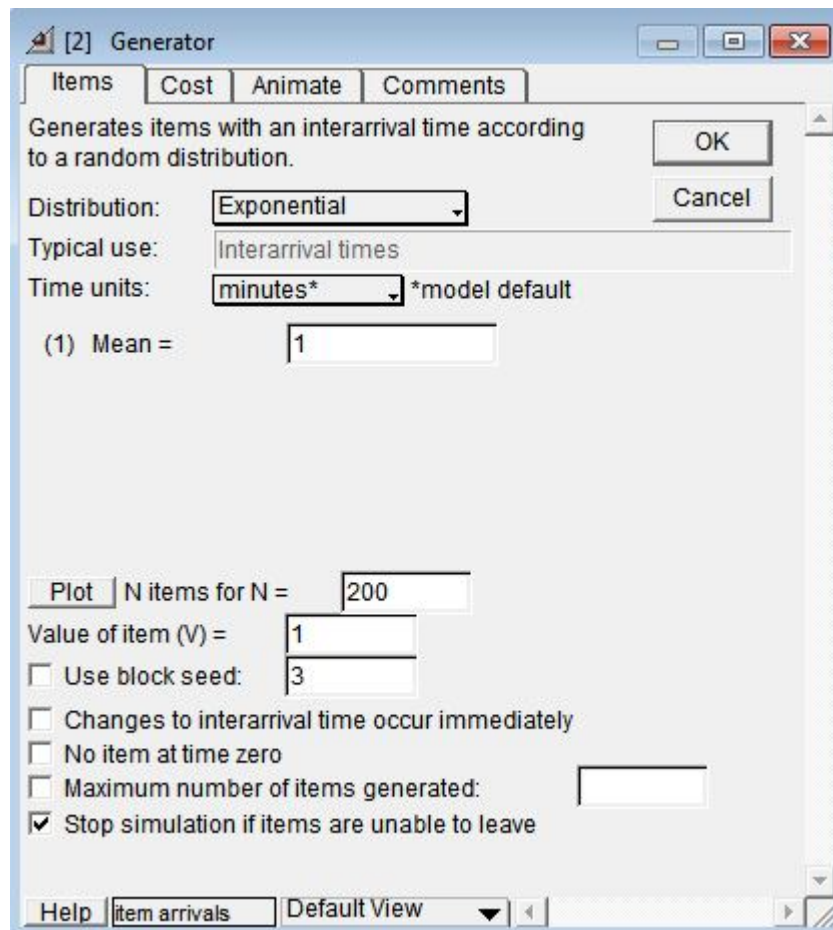
$$\text{Result} = \text{DepartureTime} - \text{ArrivalTime} ,$$

and the AGV with the lowest one is selected.

Provided that the Least IV is chosen, it is released from the queue and receives the arrived unlabeled cups. After labeling, the cups are released from the AGV and leave the hierarchical block from its output connector (Con1Out).

ii. Running the Least IV Simulation

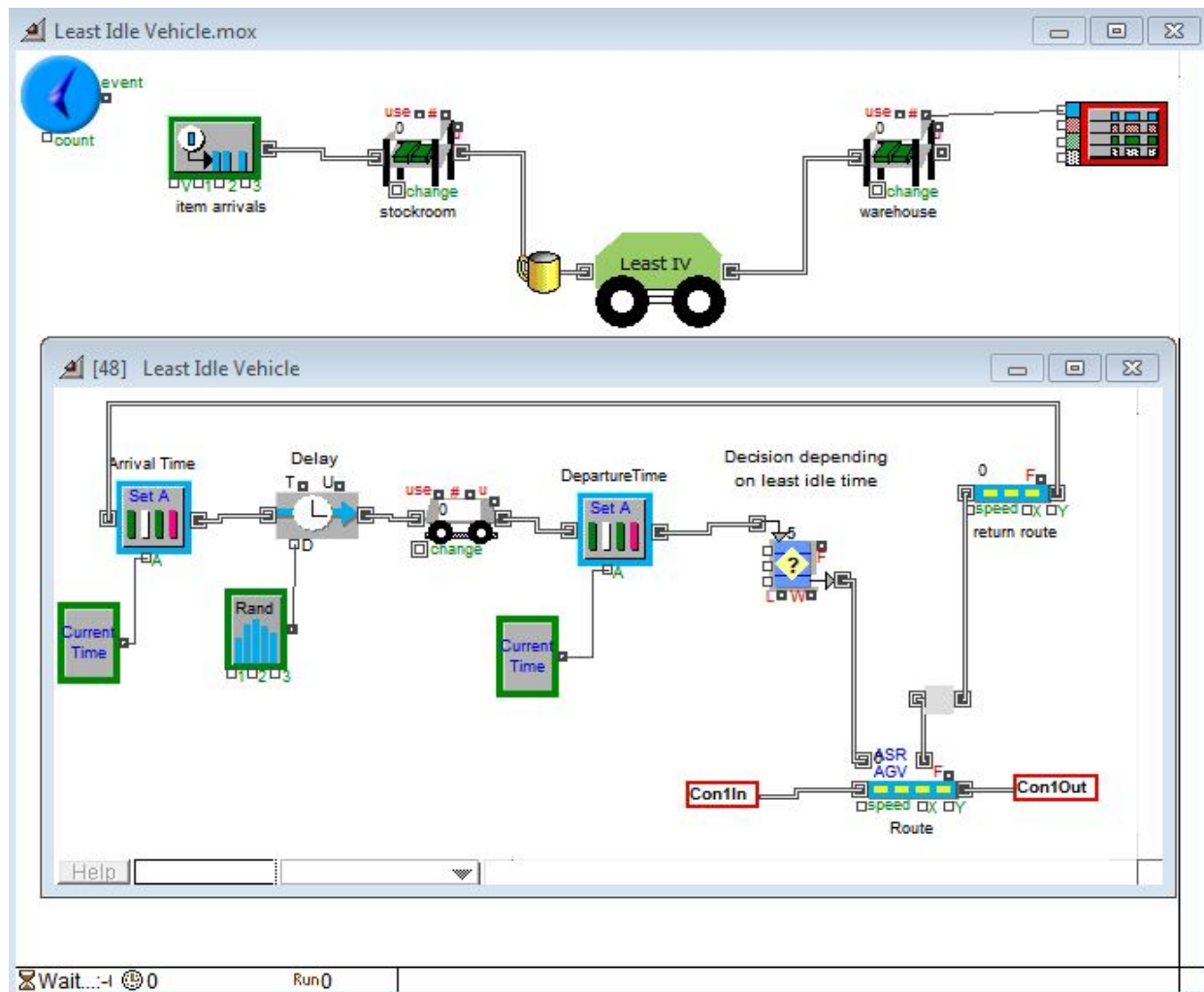
The above analysis can be fully understood by running the simulation. The simulation sets 5 available AGVs, 200 items – cups and end simulation time at 200 time units. The items arrive with the exponential distribution as it is shown in picture 3a.



Picture 3a – **The "Generator" block**

Picture 4a is a screenshot of the model at the beginning of the simulation. Under the main model there is the window which depicts the "Least IV" hierarchical block. The numbers on the blocks declare the number of items or AGVs exist in.

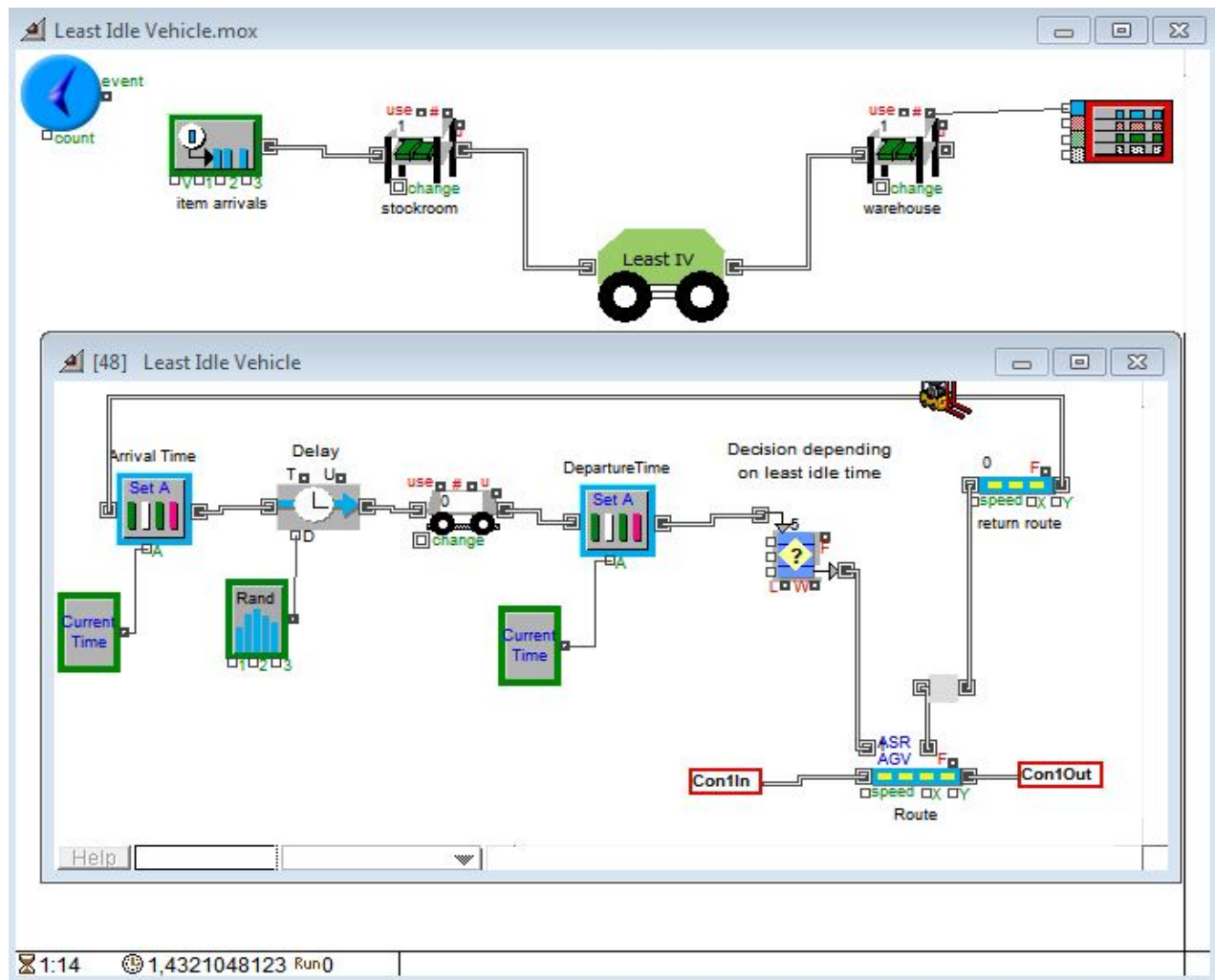
To begin with, the first cups arrive, pass through the stockroom and get into the hierarchical block from its input connector, as it is shown in the picture 4a. Simultaneously, AGVs have departed from their initial position and wait in the queue decision block. At the first simulation stage, vehicles have not completed a full cycle until they return to the "AGV" block. Consequently, the first vehicle that departed from its block is dispatched, too.



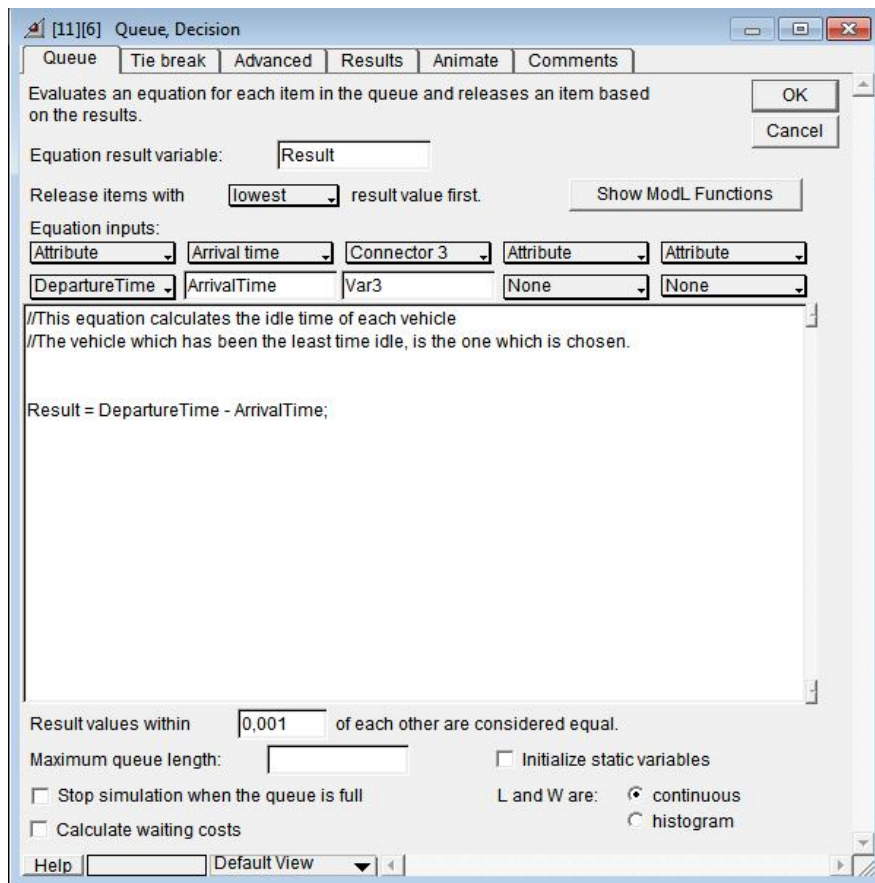
Picture 4a - **The Least IV model**

This procedure becomes more interesting after the AGVs have already returned to their block at least one time like picture 5a shows. After they have completed a full cycle, every AGV has two time attributes; arrival and departure time. These attributes are used to calculate the minimum time that the vehicles have been waiting, until an arrival occurs and begin their travel.

This calculation is made in the "Queue Decision" block. It is essential to define the desirable attributes. Otherwise the block cannot be compiled or behave properly. "Queue Decision" block's dialog is the picture 6a.

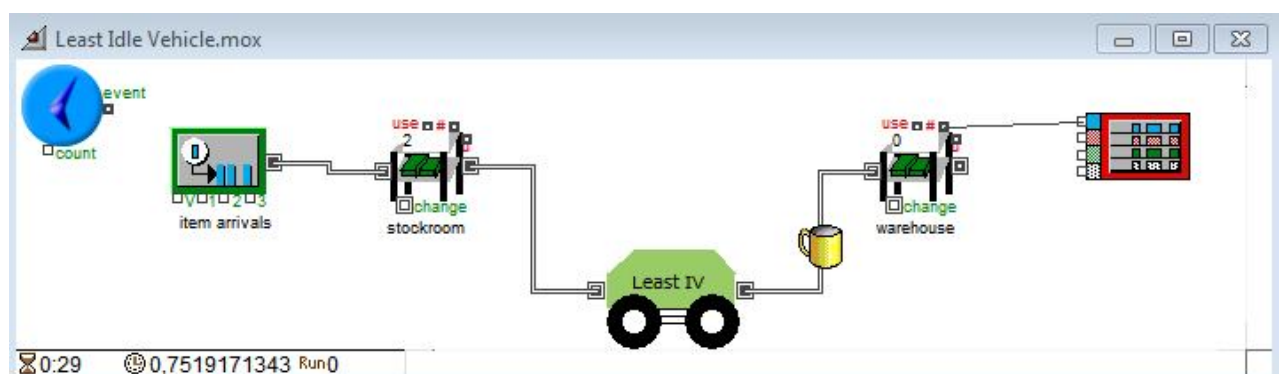


Picture 5a - *The Least IV model*



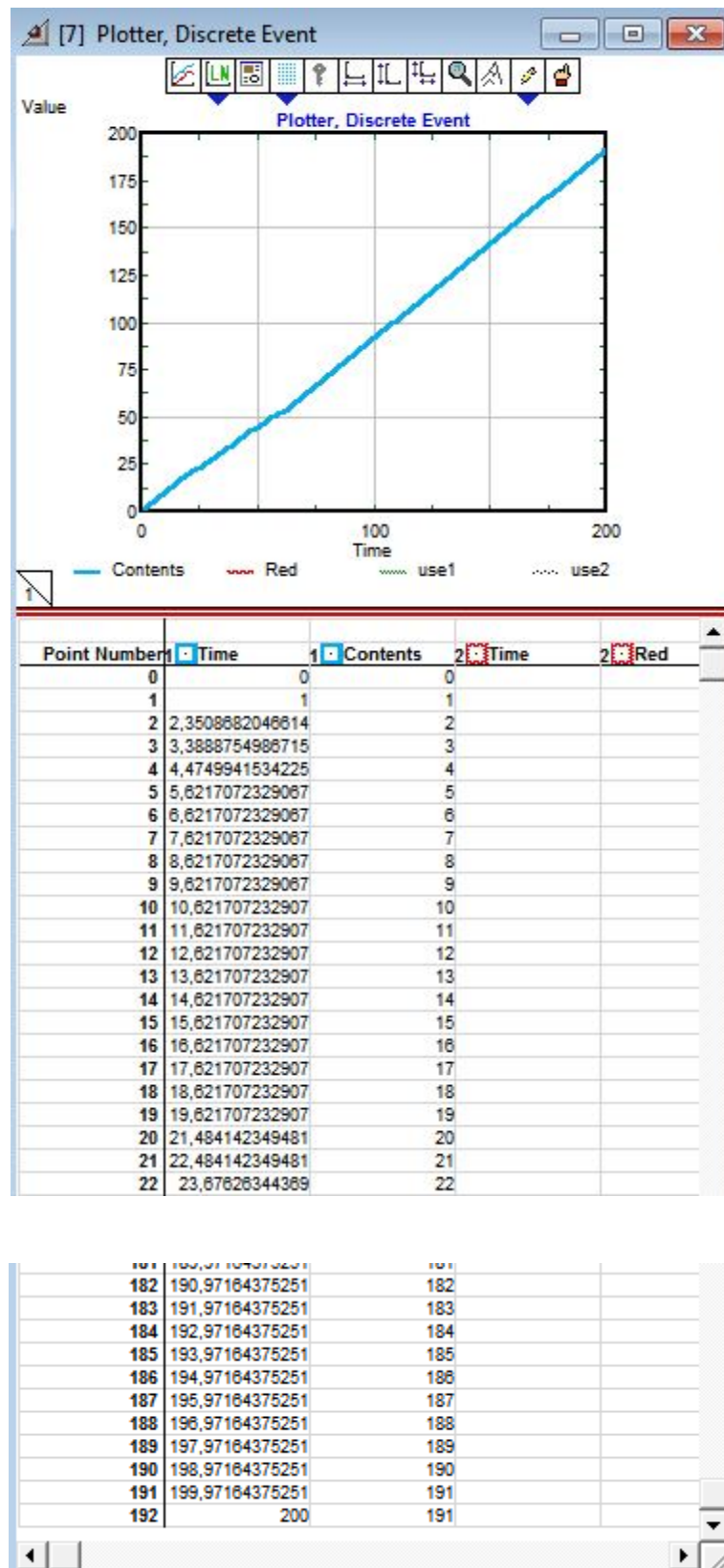
Picture 6a – **The “Queue Decision” block**

Provided that the right vehicle is dispatched and has completed its assigned job, the labeled cup leaves the hierarchical block and remains to the warehouse (Picture 7a).



Picture 7a - **The Least IV model**

The last block, the “Plotter” block is connected with the warehouse and provides a representation of the labeled cups’ arrivals.



Picture 8a - **The simulation plotter**

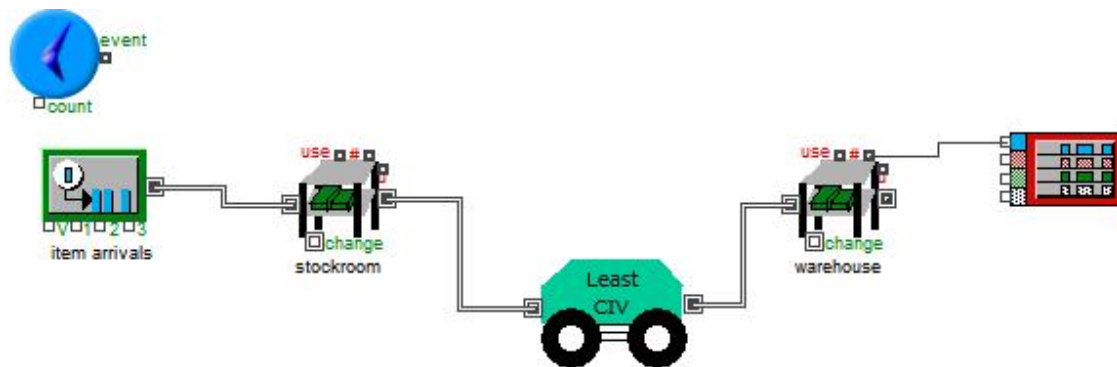
The above picture shows the relevant chart during the simulation. The “Value” axis indicates the items that passed in the warehouse, whereas the horizontal axis

indicates the units of time. Under the chart, there is a data table, as well. There are the time and the content column. Every line of the first column matches with the relevant line of the second column. The particular data table presents the first 22 values and the last 11 ones, whereas the intermediate values are omitted because of the table's magnitude.

As it can be concluded, the simulation time is in fact 200 time units and the labeled cups, which are finally ready and stocked, are 191.

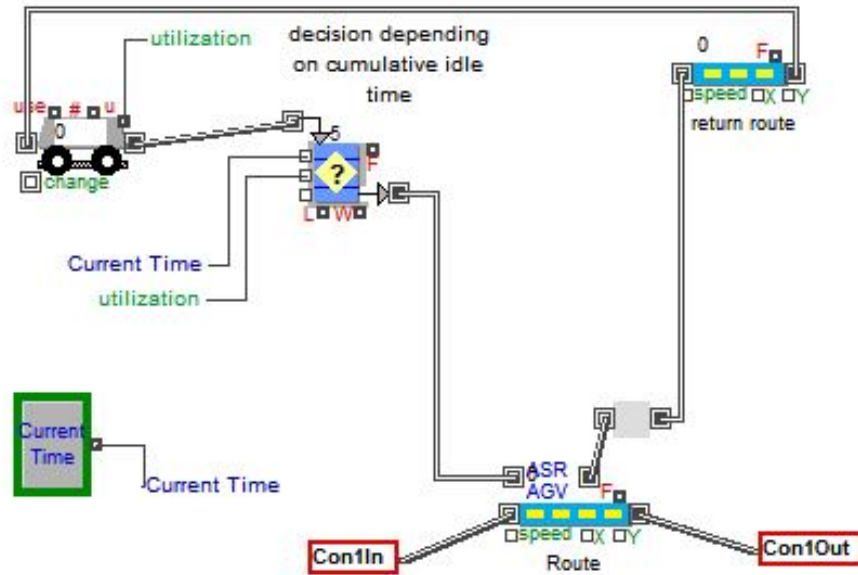
b. Least Cumulative Idle Vehicle (Least CIV)

i. About the Least CIV model



Picture 1b – **The Least CIV model**

This model (Picture 1b) calculates the entire period that vehicles have remained idle until a job accrues and have to be dispatched. If the hierarchical “Least CIV” block opens, it will be as the picture shows:



Picture 2b – The “Least CIV” hierarchical block

In this case, there are the icons “Con1In” and “Con1Out” which practically are the input and output connector of the “Least CIV” block. The unlabeled cups arrive at the conveyor belt, pass in the hierarchical block and wait for an AGV.

AGVs begin their travel from the “AGV” block and enter the “Queue Decision” block. Inside this block the cumulative idle time is calculated for each vehicle. Picture 2b shows that it is connected with two labels (“Current Time” and “Utilization”). Each label carries a value. On one hand, “Current Time” is also connected with the “System Variable” block and records the current time. On the other hand, “utilization” is also connected with the “AGV” block. The “u” connector shows the percentage of time the AGV was in use (out of the AGV block) over the course of the simulation run. Of course, this statistic only makes sense in a closed system where the AGV is cycled back to the “AGV” block.

For every incoming vehicle, “Queue Decision” block calculates the equation:

$$\text{Result} = (1 - \text{utilization}) * \text{CurrentTime};$$

The parenthesis defines the percentage of time the AGV was not in use (out of the “AGV” block) over the course of the simulation run. If this parenthesis is multiplied with current time, then the result gives the total idle time of the vehicle.

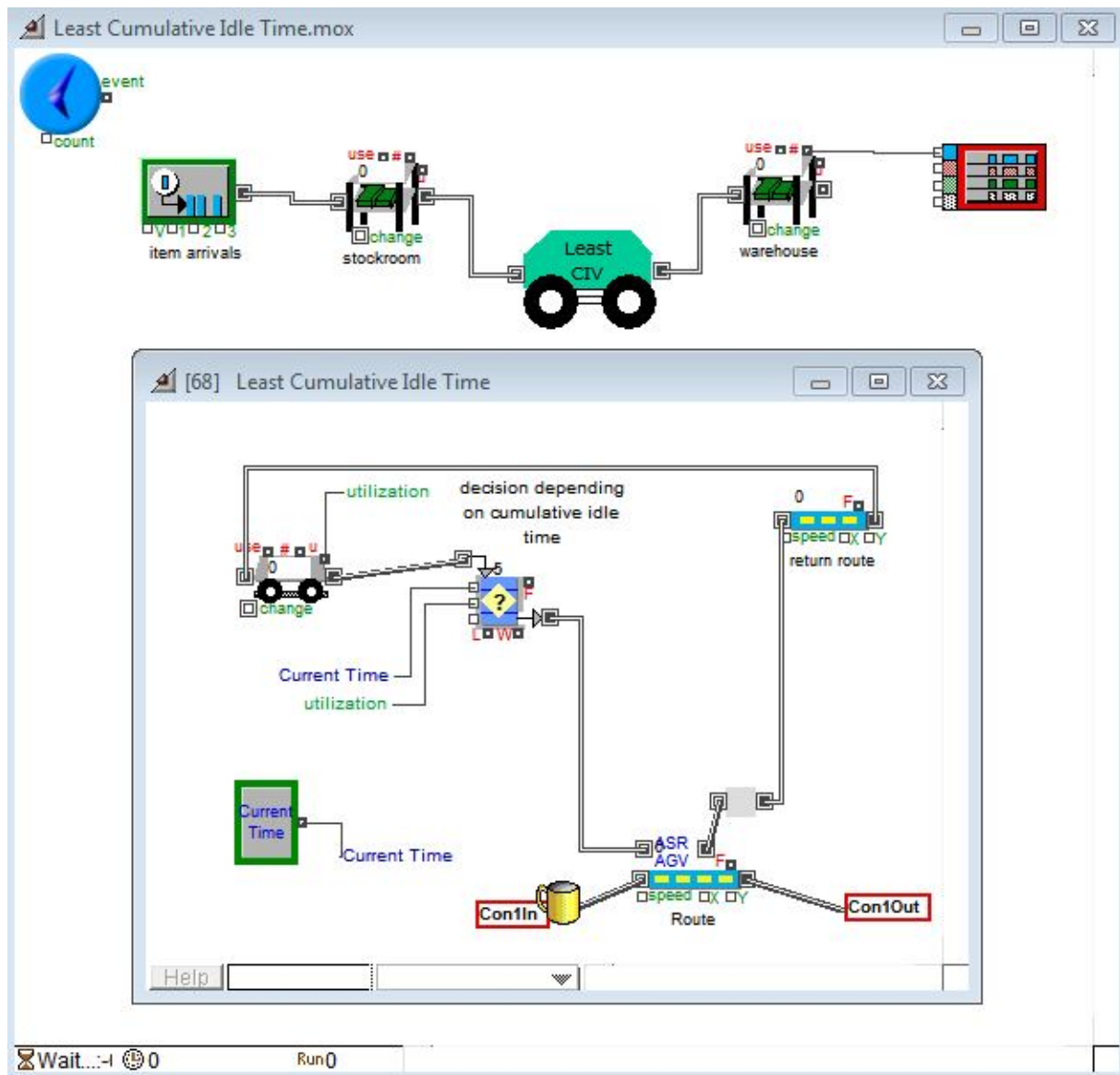
The Least Cumulative Idle Vehicle algorithm assigns the highest dispatching priority to the vehicle which has the least cumulative idle period among the others. Provided that the Least CIV is chosen, it is released from the queue and receives the

arrived unlabeled cups. After labeling, the cups are released from the AGV and leave the hierarchical block from its output connector (Con1Out).

ii. Running the Least CIV Simulation

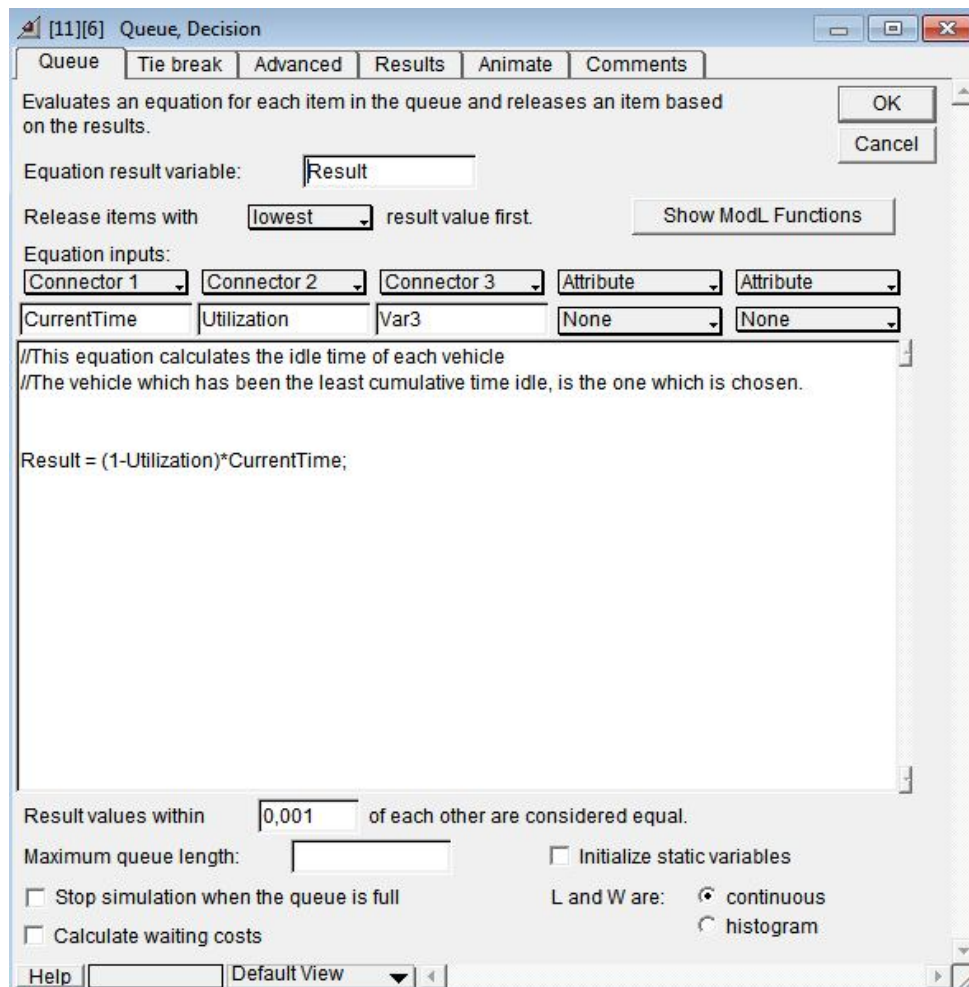
The simulation of the model can obtain a better view of the scheduling rule. The simulation has the same characteristics with the former one (5 available AGVs, 200 cups, end simulation time 200 time units).

Picture 3b is a screenshot of the model at the beginning of the simulation. This can be confirmed by the simulation time which shows 0, down on the left of the Least CIV model. Under the main model there is the window which depicts the "Least CIV" hierarchical block. Obviously, an unlabeled cup is directed towards the "Least CIV" block. At the same time, all the AGVs are in the "Queue Decision" block and wait to be dispatched, depending on their own total idle time during the simulation run.



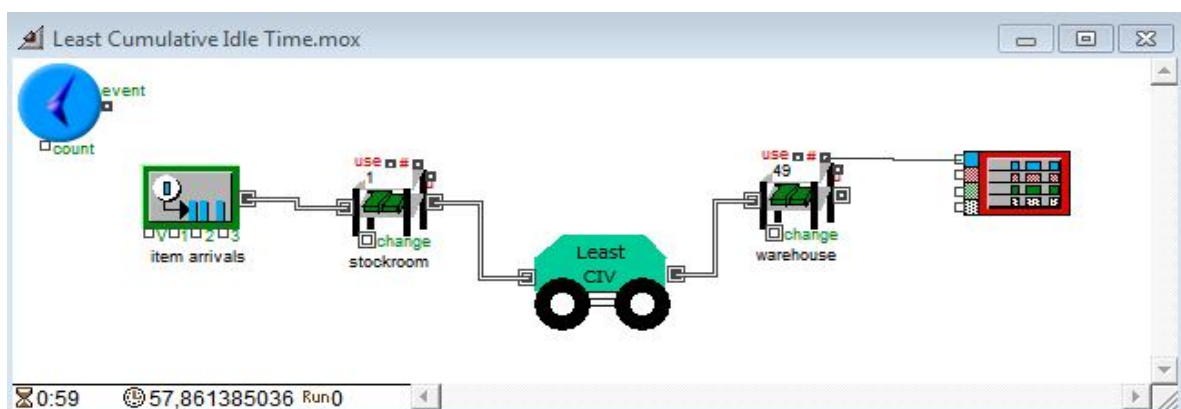
Picture 3b- **The Least CIV model**

The equation that was presented in the previous section (7.b.i) is calculated in the queue tab of the "Queue Decision" block as picture 4b shows. Contrary to the "Queue Decision" block of the Least IV model (picture 6a), this block does not use attributes, but connectors instead. Actually, that was obvious at the presentation of the subsystem in the picture above. As a result, it uses as inputs the values from the connectors (Current Time, Utilization). Thus, the equation is calculated in the tab and the algorithm selects the vehicle with the lowest result ensuring that the vehicle that will be dispatched is the one with the least cumulative idle time during the simulation.



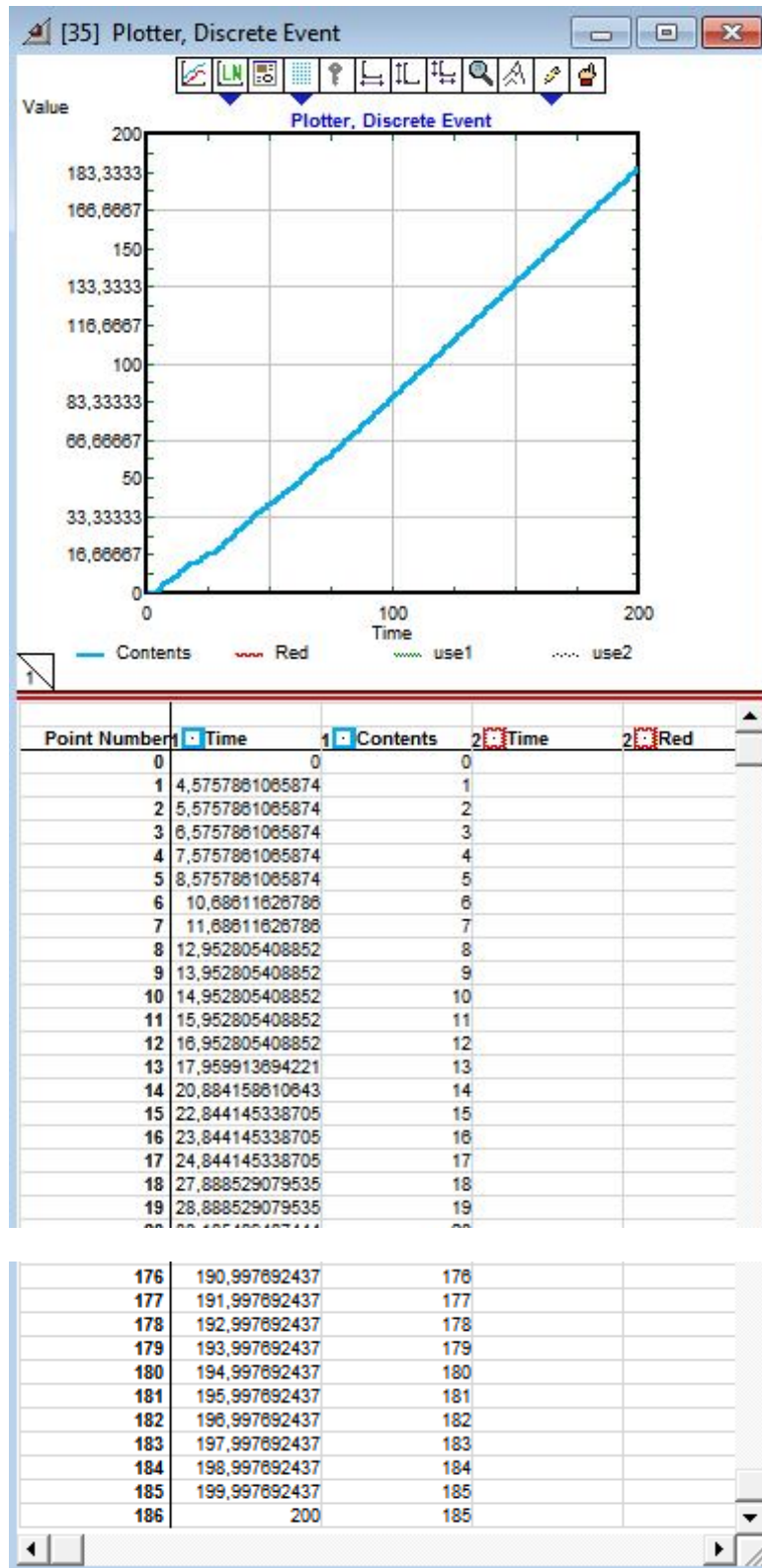
Picture 4b – The "Queue Decision" block

After the right vehicle is dispatched and has completed its assigned job, the labeled cup leaves the hierarchical block and remains to the warehouse (Picture 5b). In fact, this picture shows that when the time that has elapsed is 57,8613 units, there is 1 unlabeled cup in the stockroom and 49 labeled cups in the warehouse ready for sale.



Picture 5b - The Least CIV model

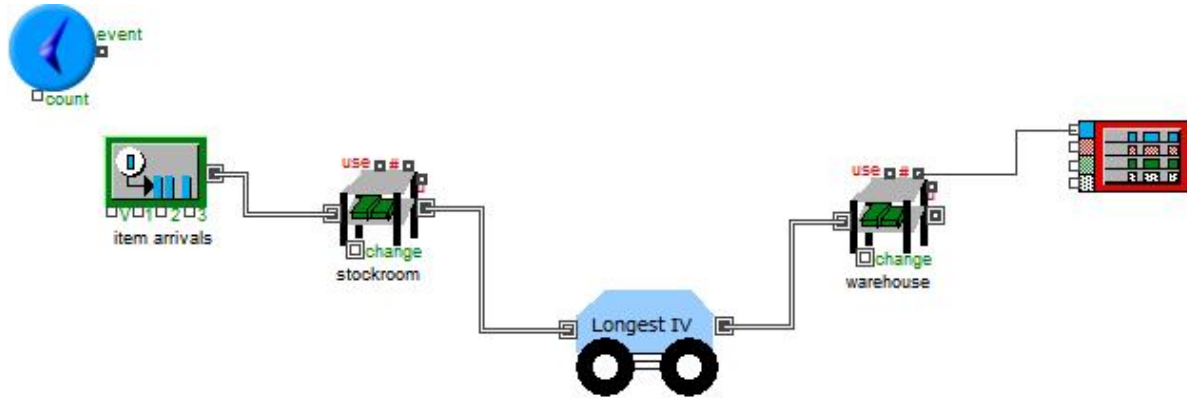
Last but not least, there is the plotter of the simulation. Finally, it seems that at the end of the simulation run (200 time units), there are 185 labeled cups in the warehouse ready for sale.



Picture 6b – *The simulation plotter*

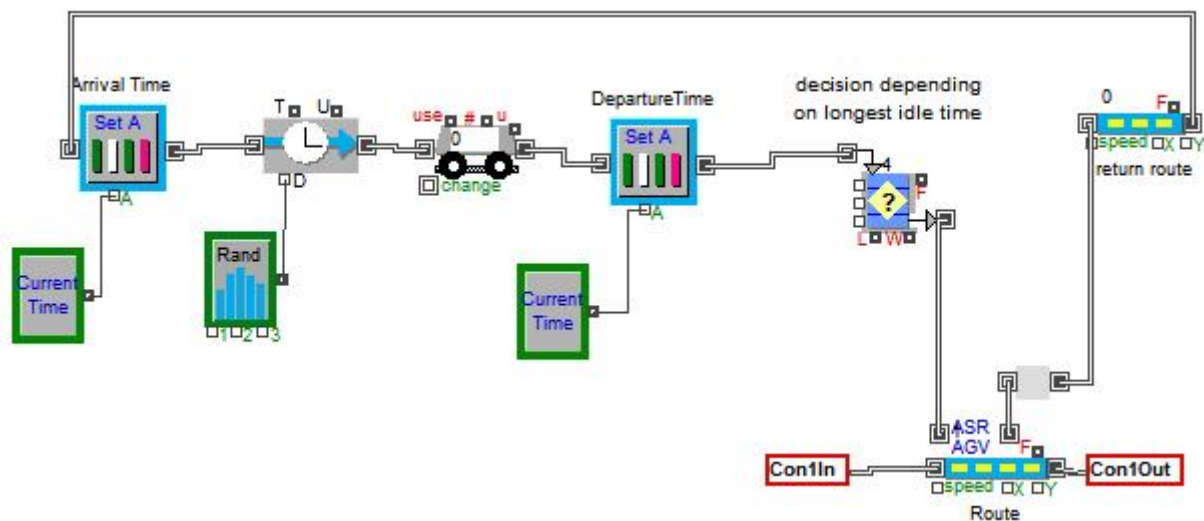
c. Longest Idle Vehicle (Longest IV)

i. About the Longest IV model



Picture 1c – **The Longest IV model**

The Longest IV model has the exact opposite objective from the Least IV. Picture 1c shows how the entire model looks like in the simulation software, whereas picture 2c depicts the subsystem, which the algorithm is applied to.



Picture 2c – **The "Longest IV" hierarchical block**

Apparently, the subsystem consists of the same blocks as the Least IV. The "Con1In" icon represents the input connector of the "Least IV" block. Respectively, the "Con1Out" icon represents the output connector of the block. Since the cups have started their route from generator block ("item arrivals" block), they pass through the "Con1In" connector and wait for an AGV.

As far as the AGV concerned, it is assumed that all the vehicles have returned at least one time to their block, so as all the vehicles have time stamps not only for their departure time, but also for their arrival.

For this reason, there are two "Set Attribute" blocks right before and after of the AGV block that record the exact time that a vehicle has when it enters or leaves the "AGV" block ("Activity Delay" block exists for representing the vehicle's delay between its entry and exit from the block). Thus, every AGV has time stamps of its arrivals and departures.

However, there is a significant difference in the queue decision block. Despite the fact that the equation is the same as formerly, the vehicle which is selected is the one with the highest result. The dispatching process is expressed by selecting a vehicle, such that the abstraction of the Arrival Time from the Departure Time gives the highest result. So, "Queue decision" block releases items according to the results of the equation defined in it. There are two different inputs that were set as parameters in this equation, Arrival and Departure Time. Thus, the result is calculated from the equation:

$$\text{Result} = \text{DepartureTime} - \text{ArrivalTime} ,$$

and the AGV with the highest one is selected.

The Longest Idle Vehicle algorithm assigns the highest dispatching priority to the vehicle which has the longest idle period among all available ones.

At the same time, an unlabeled cup has already entered the subsystem through the "ConIn1" connector and waits for an AGV. The vehicle with the longest idle time is released from the queue and is leaded to the cup. As soon as it arrives to the conveyor belt ("Route" block), receives the unlabeled cups, completes its job and releases the labeled cups, it goes back to where it started through the "return route" block. After labeling, the cups leave the hierarchical block from its output connector ("Con1Out") and are transferred to the warehouse.

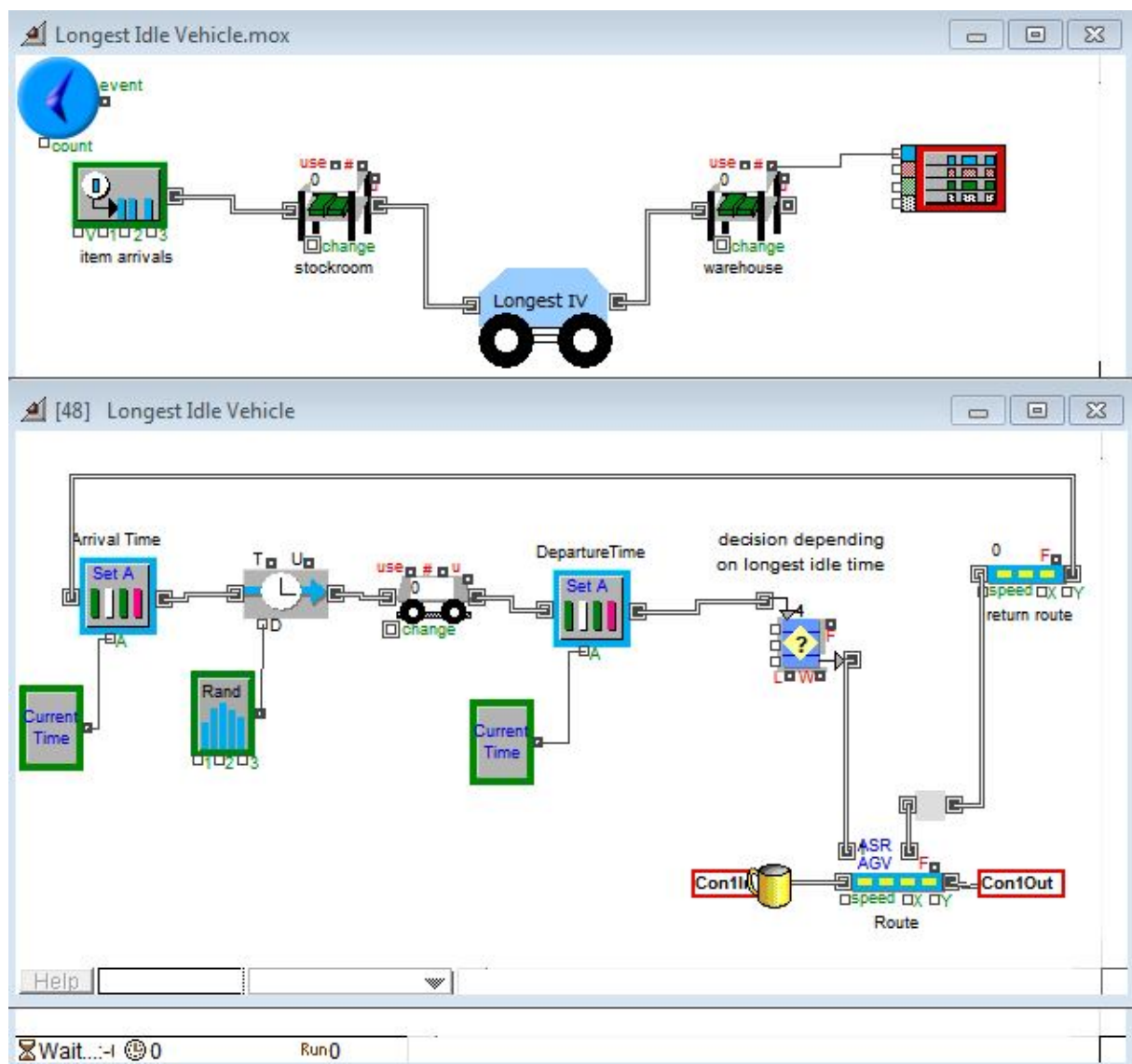
ii. Running the Longest IV Simulation

Afterwards, there is the simulation of the Longest IV model. As picture 3c shows the main model is on the top, whereas there is the inside of the hierarchical block at the bottom.

The cups are generated in the "Generator" block (item arrivals) and move to the "Longest IV" block after they have passed through the stockroom. Furthermore, by the time a cup has entered the hierarchical block and reached the "Route" block, AGVs start to move to receive it.

Picture 3c is a screenshot of the model at the beginning of the simulation. This can be seen down on the left of the picture 3c.

It is assumed that there are 5 AGVs, 200 cups and 200 time units available for the simulation run.



Picture 3c - *The Longest IV model*

As it is obvious, an unlabeled cup has entered the "Longest IV" block. At the same time, 4 AGVs are in the "Queue Decision" block and wait to be dispatched depending on their own total idle time during the simulation run.

The equation that was presented in the previous section is calculated in the queue tab of the "Queue Decision" block as picture 4c shows. Similarly to the "Queue Decision" block of the Least IV model (picture 6a), this block declares attributes (Departure Time, Arrival Time), in order to be possible for the block to be compiled.

Thus, the equation is calculated in the tab and the algorithm selects the vehicle with the highest result ensuring that the vehicle that will be dispatched is the one with the highest idle time.

[17][6] Queue, Decision

Queue Tie break Advanced Results Animate Comments

Evaluates an equation for each item in the queue and releases an item based on the results.

Equation result variable:

Release items with result value first.

Equation inputs:

<input type="text" value="Attribute"/>	<input type="text" value="Arrival time"/>	<input type="text" value="Connector 3"/>	<input type="text" value="Attribute"/>	<input type="text" value="Attribute"/>
<input type="text" value="DepartureTime"/>	<input type="text" value="ArrivalTime"/>	<input type="text" value="Var3"/>	<input type="text" value="None"/>	<input type="text" value="None"/>

```
//This equation calculates the idle time of each vehicle
//The vehicle which has been the most time idle, is the one which is chosen.

Result = DepartureTime - ArrivalTime;
```

Result values within of each other are considered equal.

Maximum queue length:

☐ Initialize static variables

☐ Stop simulation when the queue is full

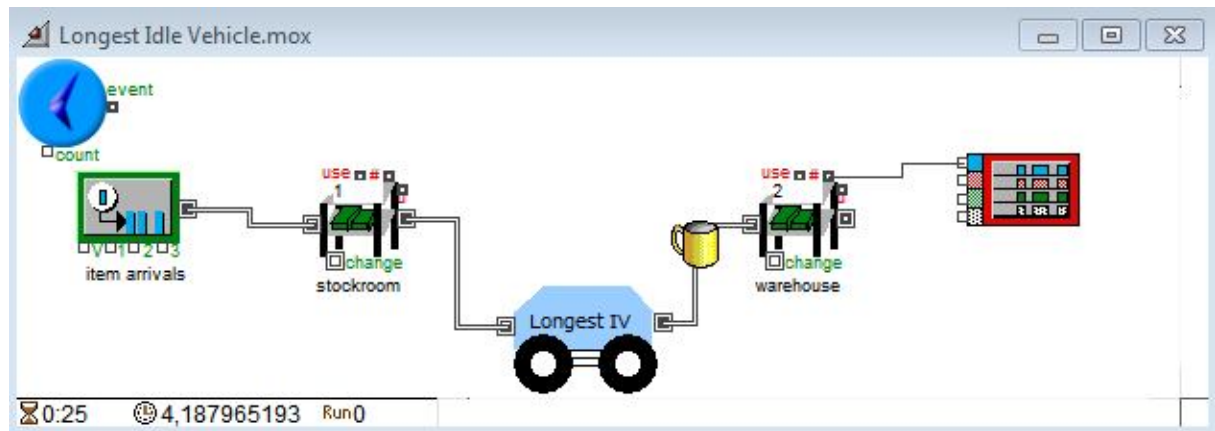
L and W are: ☒ continuous ☐ histogram

☐ Calculate waiting costs

Help Default View

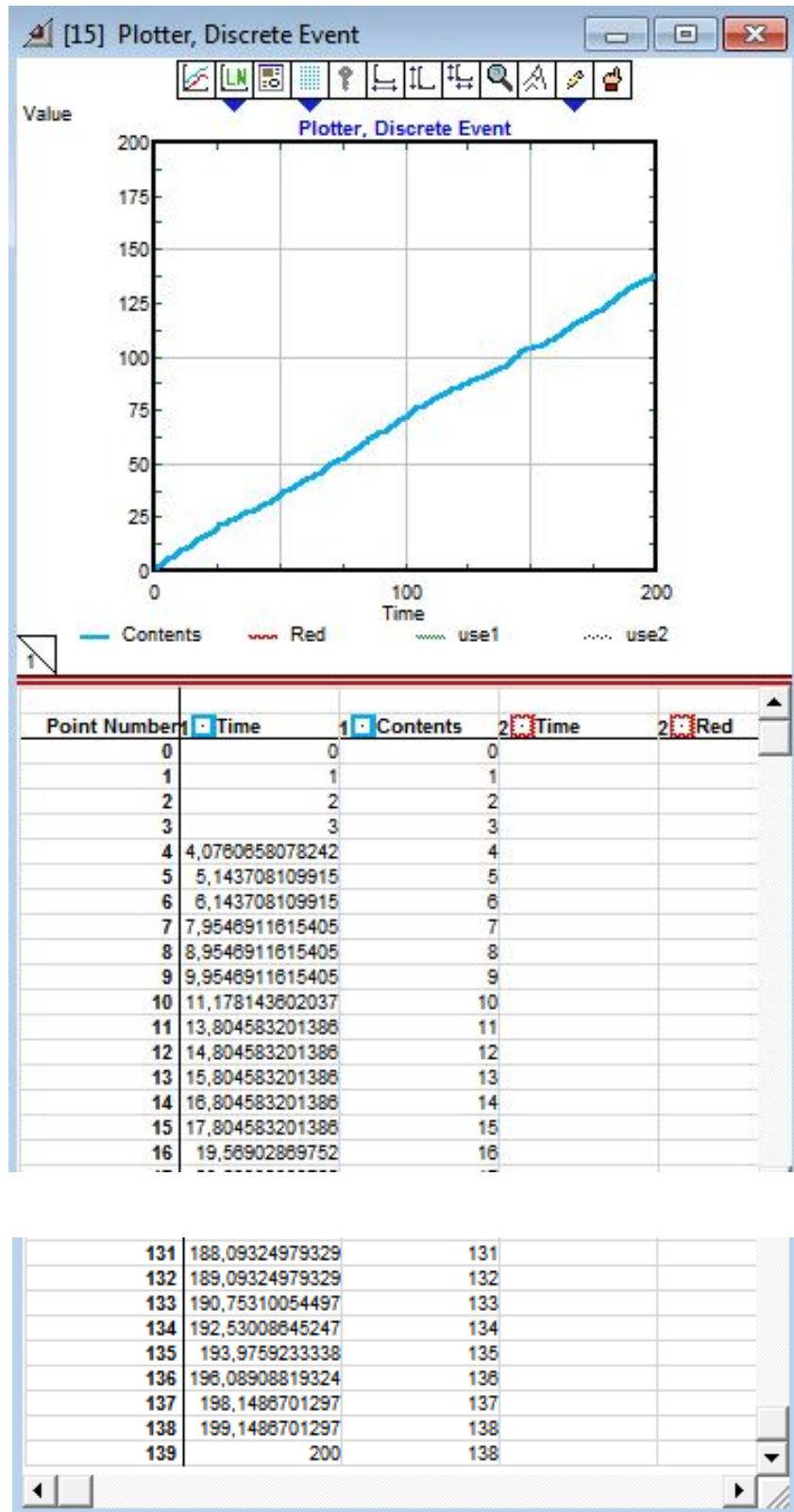
Picture 4c – The "Queue Decision" block

Provided that the appropriate vehicle is dispatched and has completed its assigned job, the labeled cup leaves the hierarchical block and remains to the warehouse (Picture 5c).



Picture 5c - The Longest IV model

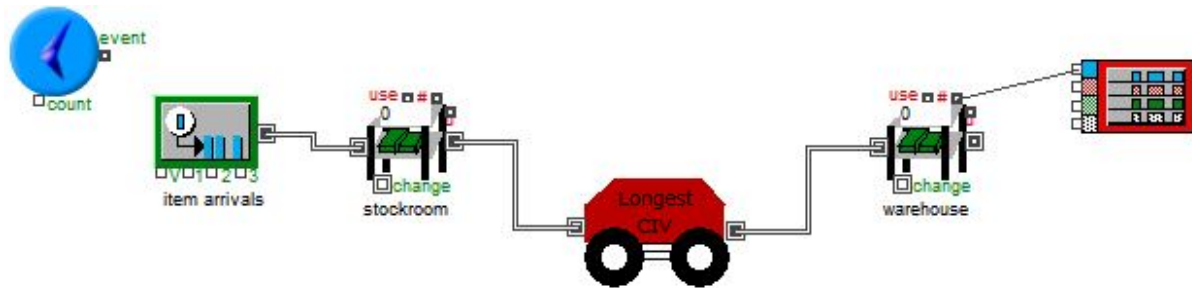
Finally, there is the plotter of the simulation. For every time there is the amount of the labeled cups that pass in the warehouse. It seems that at the end of the simulation run (200 time units), there are 138 labeled cups in the warehouse ready for sale.



Picture 6c – The simulation plotter

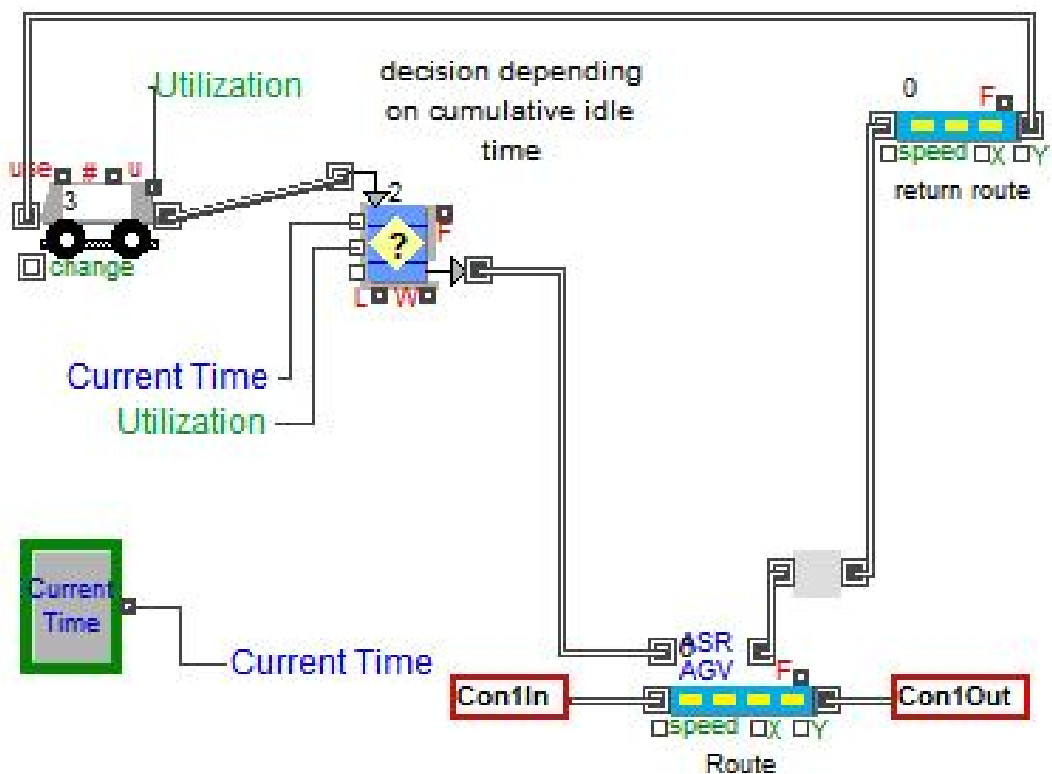
d. Longest Cumulative Idle Vehicle (Longest CIV)

i. About the Longest CIV model



Picture 1d - **The Longest CIV model**

Longest CIV model (Picture 1d) calculates the entire period that vehicles have remained idle until a job accrues and have to be dispatched. This case is the antithesis of the Least CIV model. If the hierarchical "Longest CIV" block opens, it will be as the picture shows:



Picture 2d – **The "Longest CIV" hierarchical block**

Once again, there are the input and output connector of the "Longest CIV" block which are represented by the "Con1In" and "Con1Out" icons. The unlabeled cups arrive at the conveyor belt, pass in the hierarchical block and wait for an AGV.

In the following, AGVs begin their travel from the "AGV" block and enter the "Queue Decision" block. Inside this block the cumulative idle time is calculated for each vehicle. Picture 2b shows that it is connected with two labels ("Current Time" and "Utilization"). Each label carries a value.

On one hand, "Current Time", apart from "Queue Decision" block, is also connected with the "System Variable" block and records the current time. On the other hand, "Utilization" is also connected with the "AGV" block. The "u" connector shows the percentage of time the AGV was in use (out of the AGV block) over the course of the simulation run. Of course, this statistic only makes sense in a closed system where the AGV is cycled back to the "AGV" block.

For every incoming vehicle, "Queue Decision" block calculates the equation:

$$\text{Result} = (1 - \text{utilization}) * \text{CurrentTime};$$

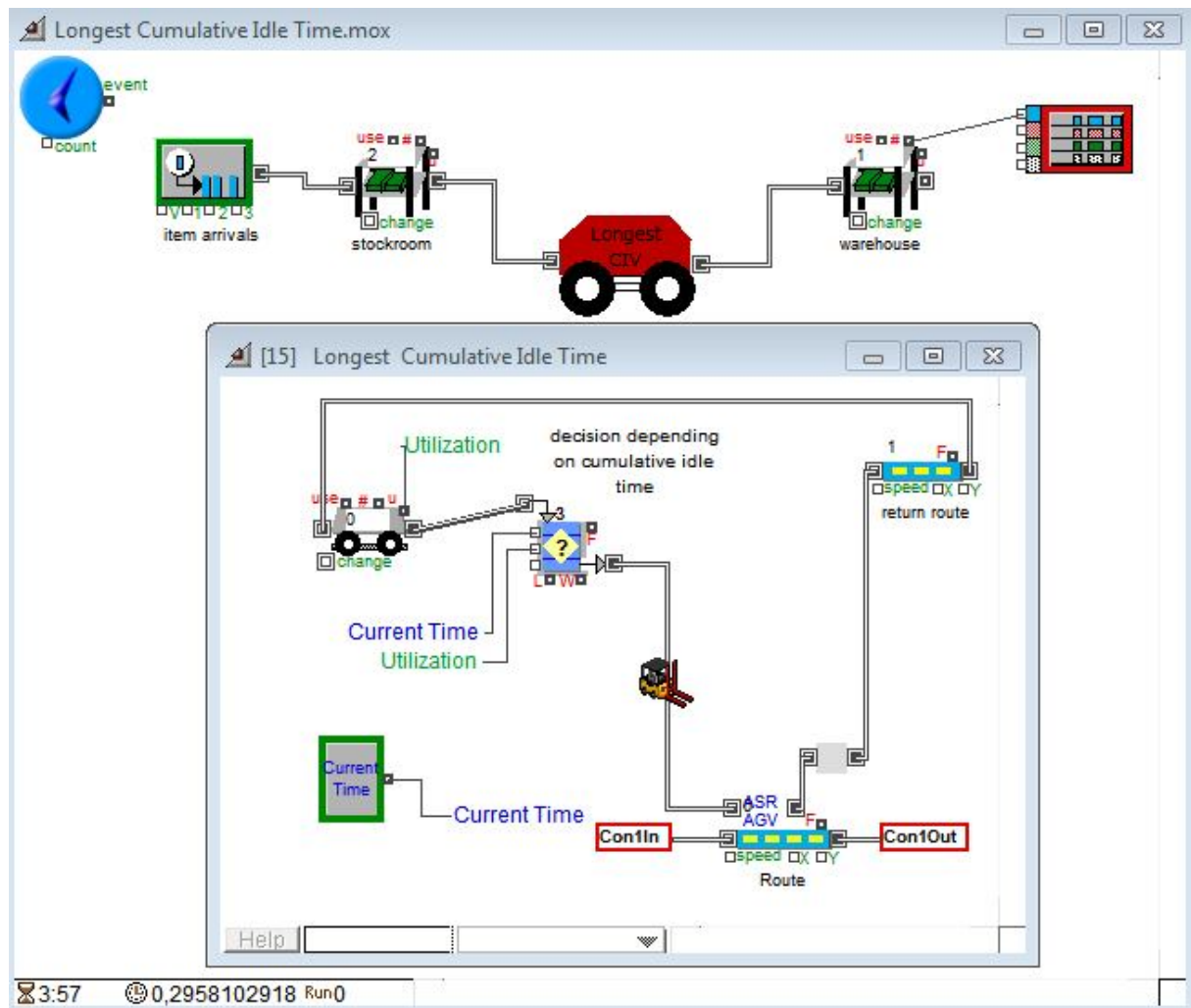
The parenthesis is known that defines the percentage of time the AGV was not in use (out of the AGV block) over the course of the simulation run. If this parenthesis is multiplied with current time, then the result gives the total idle time of the vehicle.

The Longest Cumulative Idle Vehicle algorithm assigns the highest dispatching priority to the vehicle which has the longest cumulative idle period among the others. Provided that the Longest CIV is chosen, it is released from the queue and receives the arrived unlabeled cups. After labeling, the cups are released from the AGV and leave the hierarchical block from its output connector ("Con1Out").

ii. Running the Longest CIV model

The following section contains an example of how actually works the Longest CIV model. There are 5 AGVs, 200 unlabeled cups and 200 time units available. According to picture 3d, the elapsed time of simulation is 0,2958 time units, yet the remaining real time is 3 minutes and 57 seconds. At this moment, there is a cup arrival in the "Longest CIV" block, which needs an AGV to run through the "Route" block.

There are already 3 AGVs in the "Queue Decision" block (as it is shown in the picture), but only one will serve. This vehicle has to be the one with the longest cumulative idle time during the simulation run.



Picture 3d – **The Longest CIV model**

This decision is made inside this block. More specifically, there is an equation that is calculated every time a vehicle passes through. This equation is the one which was presented in section 7.d.i. Thereafter, the block is formed as Picture 4d shows. According to this screenshot, there are no declared attributes but connectors instead. So, there is Connector1 which is connected with CurrentTime and Connector2 which is connected with Utilization. So, the inputs of the equation use the connectors' values.

As far as the equation is stated, and the result for every vehicle is calculated, the AGV with the highest one is released.

[11][6] Queue, Decision

Queue Tie break Advanced Results Animate Comments

Evaluates an equation for each item in the queue and releases an item based on the results.

Equation result variable:

Release items with result value first.

Equation inputs:

<input type="text" value="Connector 1"/>	<input type="text" value="Connector 2"/>	<input type="text" value="Connector 3"/>	<input type="text" value="Attribute"/>	<input type="text" value="Attribute"/>
<input type="text" value="CurrentTime"/>	<input type="text" value="Utilization"/>	<input type="text" value="Var3"/>	<input type="text" value="None"/>	<input type="text" value="None"/>

//This equation calculates the idle time of each vehicle
 //The vehicle which has been the most time idle, is the one which is chosen.

Result = (1-Utilization)*CurrentTime;

Result values within of each other are considered equal.

Maximum queue length:

☐ Initialize static variables

☐ Stop simulation when the queue is full

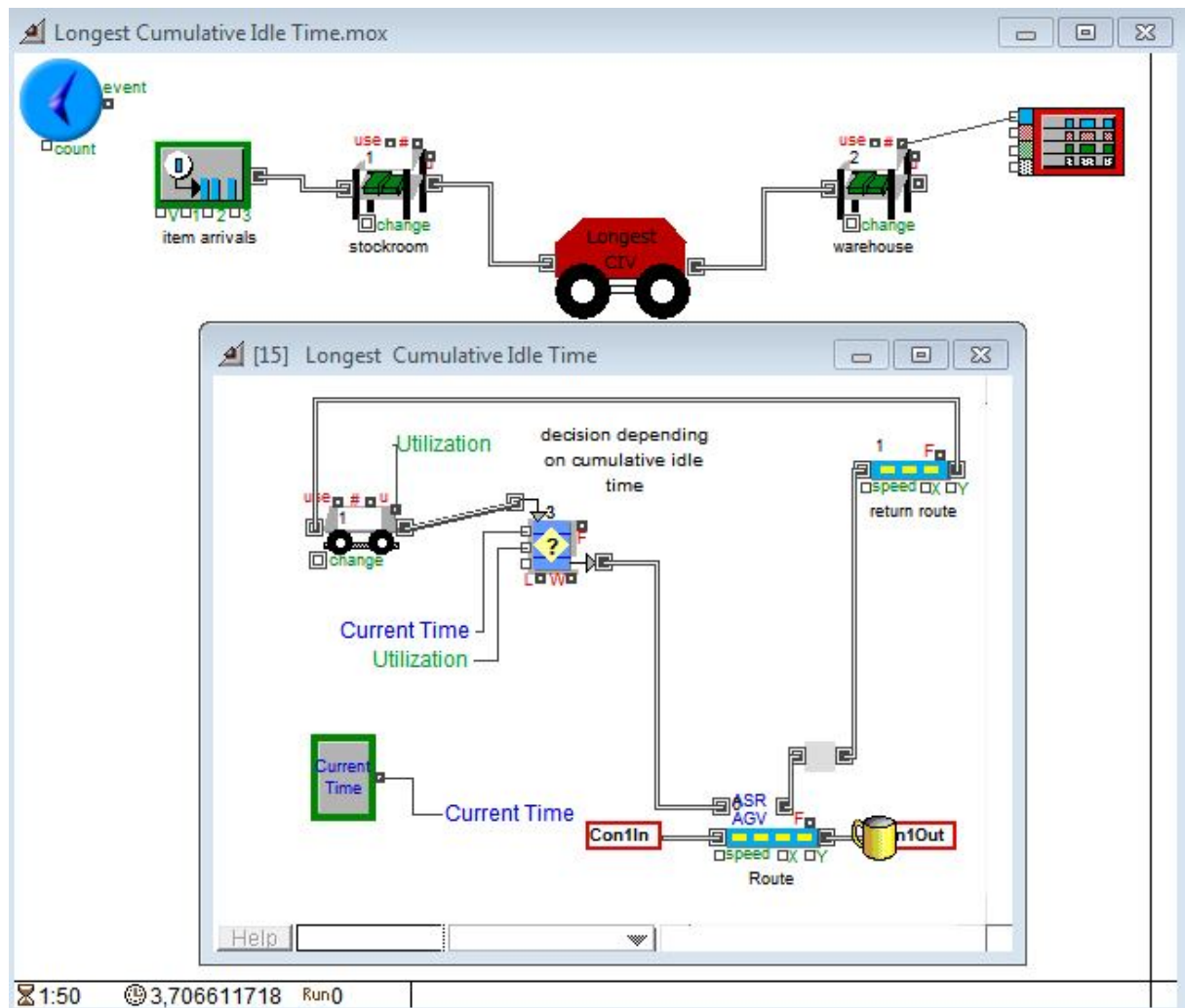
☐ Calculate waiting costs

L and W are: ☒ continuous ☐ histogram

Help Default View

Picture 4d – **The "Queue Decision" block**

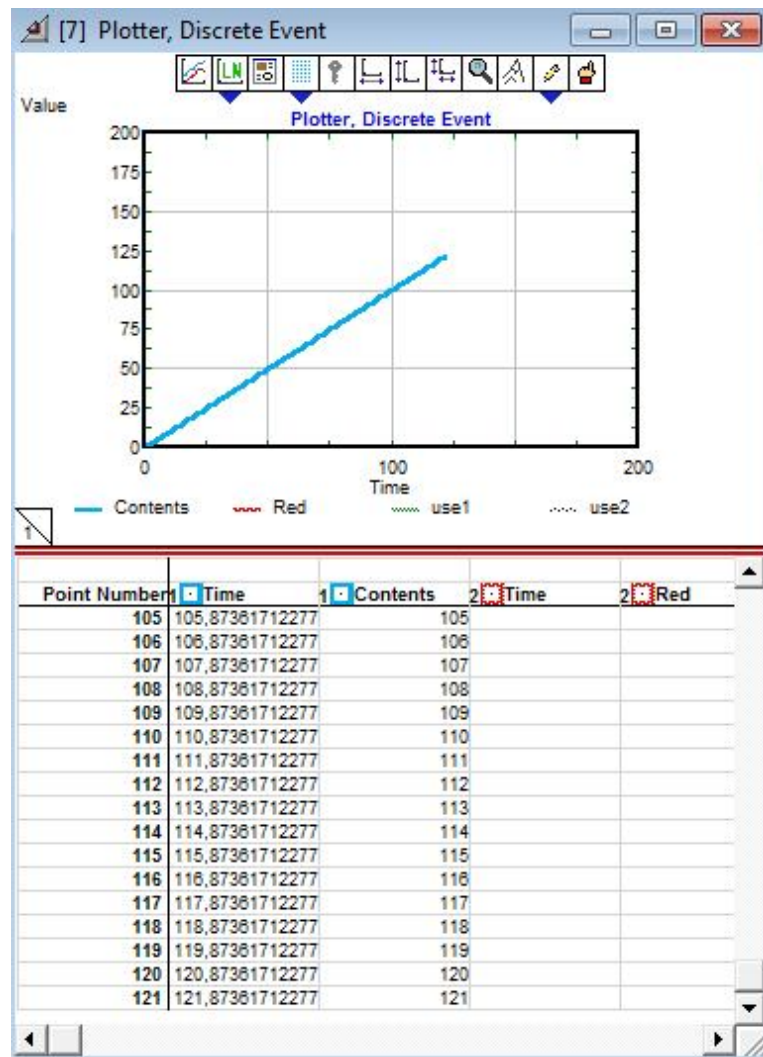
After the right vehicle is dispatched and has completed its assigned job, the labeled cup leaves the hierarchical block and remains to the warehouse (Picture 5d).



Picture 5d – **The Longest CIV model**

At last it is essential to quote the diagram of the “Plotter” block that is connected with the warehouse. This diagram denotes the moment that each cup passed in the warehouse after the process of labeling.

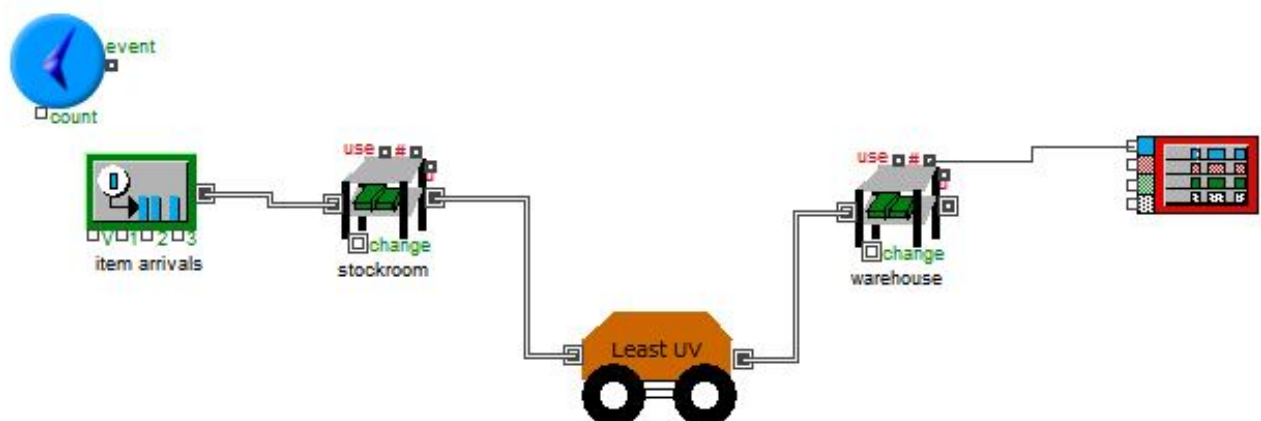
The picture below provides the exact time that each item (cup) got into the warehouse. For example, when the time is 121,873617 minutes, the warehouse accepts the 121st labeled cup. These data are given both with a diagram and with a data table lying underneath.



Picture 6d – The simulation plotter

e. Least Utilized Vehicle (Least UV)

i. About the Least UV model



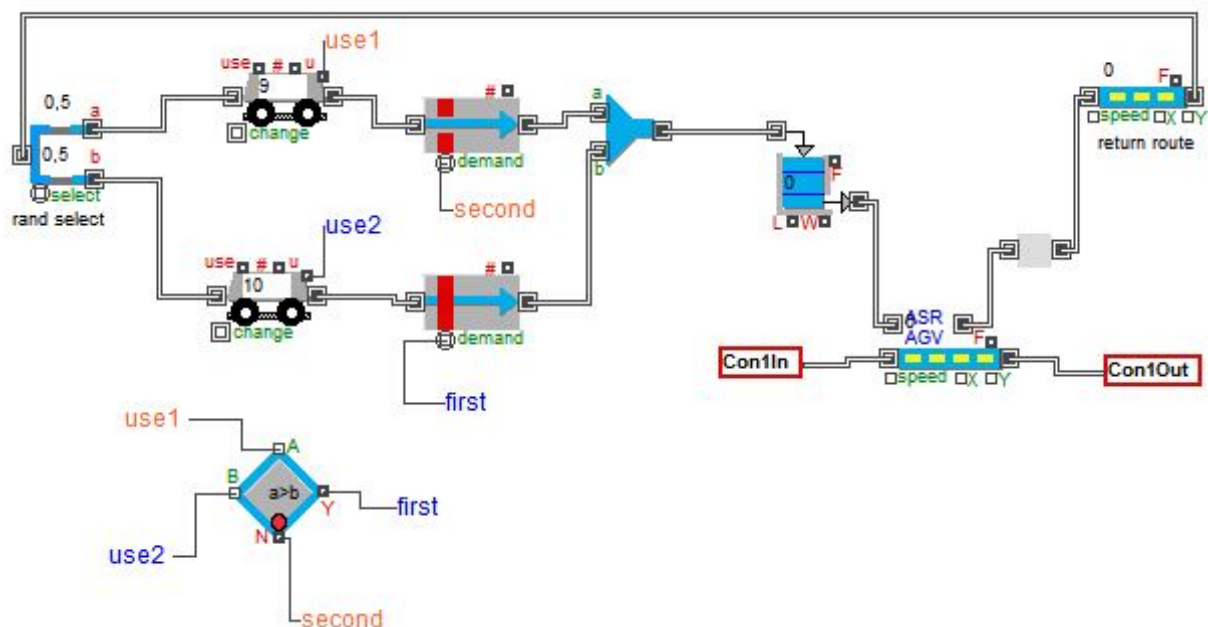
Picture 1e - **The Least UV model**

Least Utilized Vehicle model simulates the labeling operation by using a rule that ensures uniform work load on the vehicles circulating in the system. Briefly, this algorithm records the utilizations of every vehicle and releases the vehicle which has the smallest one.

In detail, picture 1e shows the Least Utilized Vehicle model which primarily consists of the "Executive" block, the core of the simulation. Thereafter, there is the "Generator" block which generates the cups' arrivals and two "Stock" blocks, one of which is connected with a "Plotter". Finally there is a hierarchical block, which is in fact, the heart of the describing rule.

Below, there is the depiction of the "Least UV" block (Picture 2e). To begin with, there are 2 "AGV" blocks. Both blocks have a "u" connector. This connector measures the utilization of each vehicle that is, the percentage of time the AGV was in use (out of the AGV block) over the course of the simulation run.

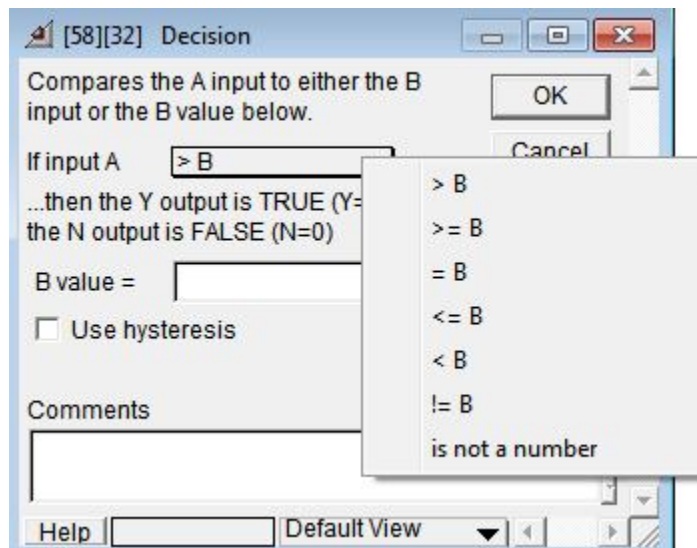
This connector of the first "AGV" block from the top is connected with a label ("use1"). This label is also connected with the "A" connector of the "Decision" block. This block lies separately under the sub model. The "u" connector of the second "AGV" block is connected with another label ("use2"). Respectively, "use2" is connected with the "b" connector of the "Decision" block. However it is essential to amplify how exactly "Decision" block operates.



Picture 2e – **"The Least UV" hierarchical block**

This block makes a decision based on the inputs and the defined internal logic. Just as picture 3e illustrates, the dialog performs the following tests comparing A to B: greater than, greater than or equal to, equal to, less than, less than or equal to, and not equal. There is also a test for A being an invalid number (noValue). The block compares the two inputs (A and B). If only the A input is connected, the block compares that value to the B value in the dialog. If the selected test is true, the Y connector (first) outputs 1 and the N connector (second) outputs 0. If the test is false, the Y connector outputs 0 and the N connector outputs 1.

As far as Least UV rule concerned, there is a specific condition under which the appropriate vehicle is chosen; if input A ("use1") is greater than input B ("use2"), it releases the vehicle which is connected with the use2 label and vice versa.



Picture 3e – The “Decision” block

Since the comparison between the utilizations is made and the least utilized vehicle is released, it passes through the “Activity Delay” block. This block operates as a gate and lets an item pass when the demand connector is connected and certain conditions exist at the demand input (if the vehicle is the least utilized one).

As follows, Least UV exits from the “Activity Delay” block and enters the “Combine” block which leads the vehicle to a queue. This a simple queue according to which, the first AGV that comes, the first leaves, too. Unlike the other described models, the heart of the algorithm is not in this block; its role is quite neutral.

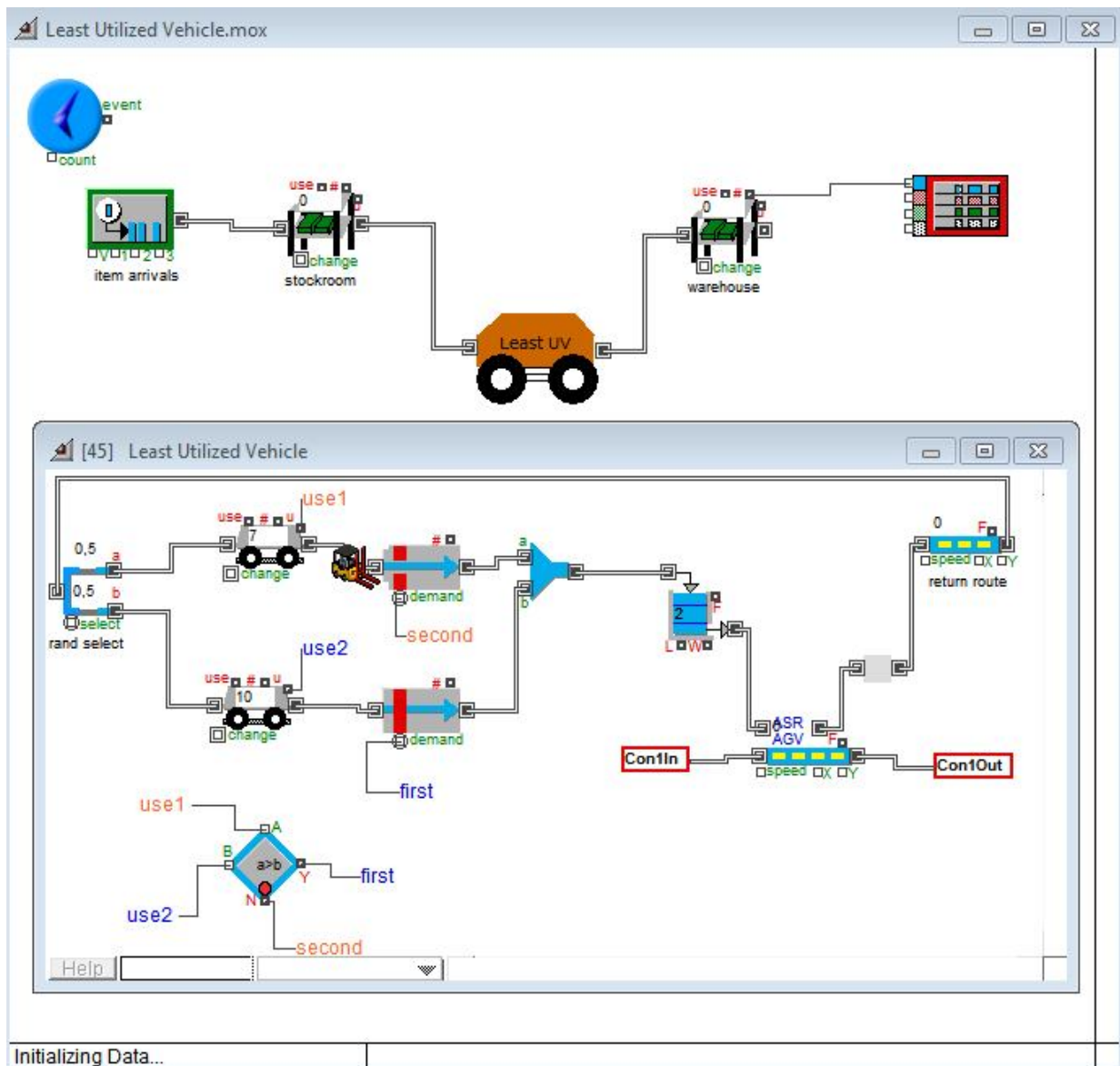
Hence, the vehicle passes through it, and enters the “Route” block waiting for an unlabeled cup. From the moment AGV completes its assigned job, it can release the labeled cup and return to where it started.

On the way to the start vehicles pass through the "Select output" block. As it is clear from picture 2e, the rates on the fork declare that the incoming AGVs have the same possibility to go either from the top or from the bottom.

ii. Running the Least UV model

Let's assume that there are 200 unlabeled cups, 20 available AGVs and 200 time units. Furthermore, there are two separate types of AGVs have of which are connected with "use1" label and the other half are connected with "use2" label (Picture 4e).

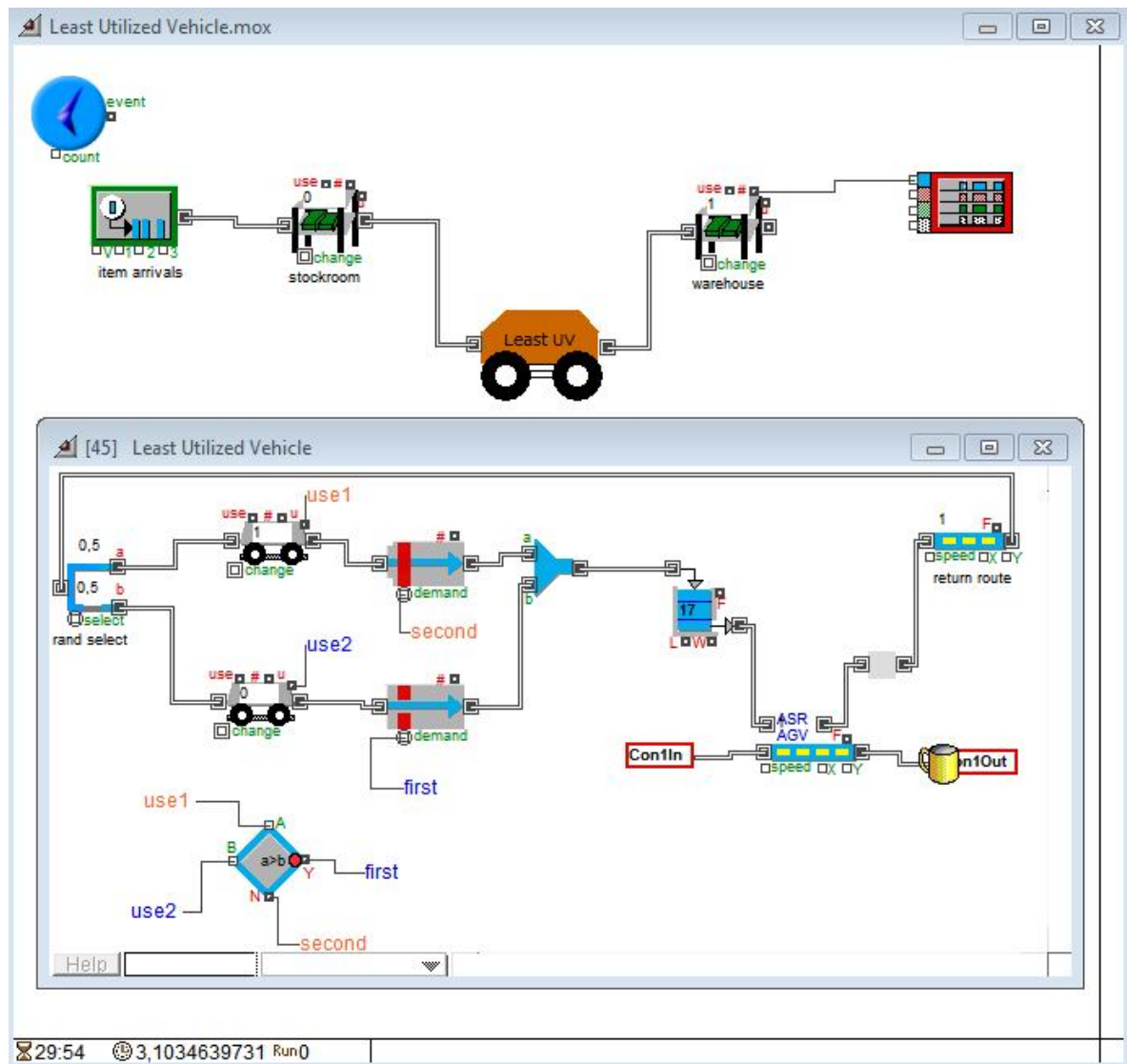
Suppose that two vehicles start to move, one from the top block and the other from the bottom. "Decision" block compares constantly the utilizations of the vehicles and chooses the appropriate depending on which utilization is the smallest.



Picture 4e – **The Least UV model**

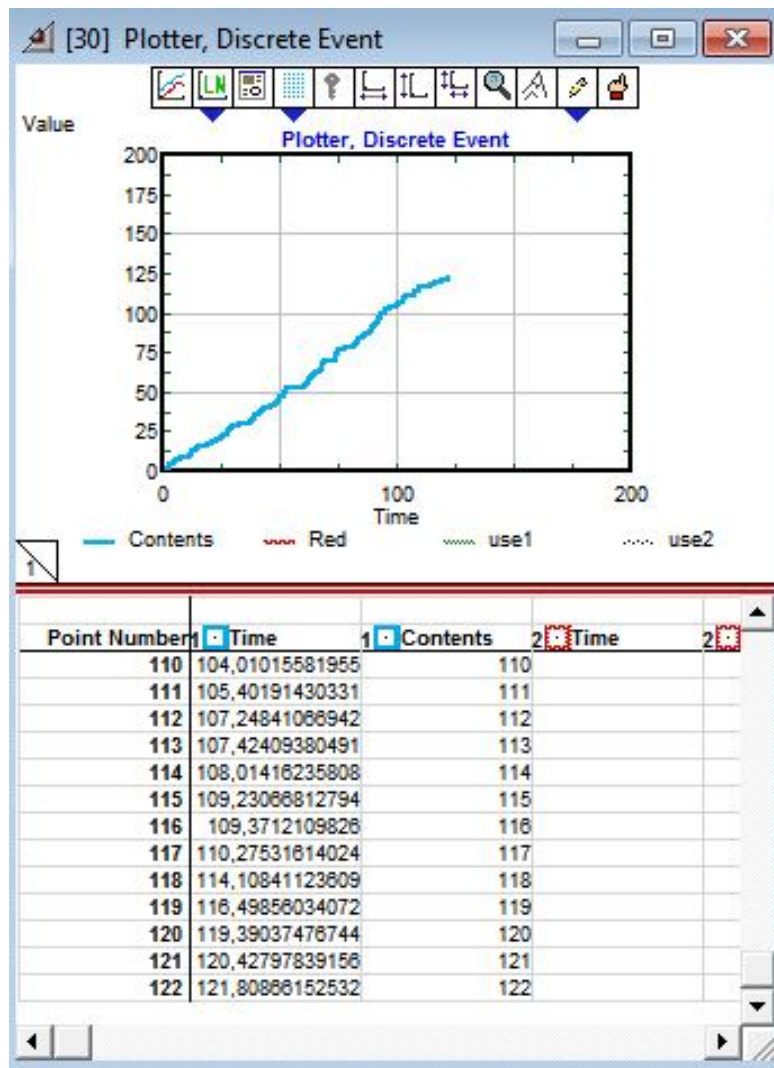
Provided that the least utilized vehicle is released, it is directed to the first-in-first-out queue. At this point, there has been already a cup arrival. This unlabeled cup is at the "Route" block waiting for an AGV to be transferred. Thus, the released AGV takes the cup and transfer it to another place for labeling. The entire process takes place in the route block.

Picture 5e is a screenshot of the simulation the moment that the labeling process is finished, the labeled cup is released and is moving to the warehouse. The elapsed time is 3,10346 time units as it is also shown underneath.



Picture 5e – **The Least UV model**

A diagram provides a good depiction of the cup arrivals during the simulation run. So, when the time is 0 there are no labeled cups in the warehouse. Similarly, the time that the screenshot is taken (121,80866 time units), there are already 122 cups in the warehouse.



Picture 6e – *The simulation plotter*

f. Nearest Vehicle (NV)

In a system using the NV rule, the load or work-center has the dispatching initiative. When a load enters the system, it places a move request; the shortest distance along the traveling paths to every available vehicle is then calculated. The idle vehicle, whose travel distance is the shortest, is dispatched to the point of request.

However, there is a difficulty in how to measure the distance along the paths. Considering how a manufacturing system works, there have been proposed two different ways to measure the distance and decide the dispatching vehicle according to given data. Thus, under the Nearest Vehicle model there are two sub models; Nearest Vehicle (distance) and Nearest Vehicle (speed).

Nearest Vehicle (distance) model assumes that the AGVs are placed in the system in given, fixed positions. Subsequently, the problem of which vehicle to dispatch is quite easy. All the given distances are collected and the vehicle with the shortest is chosen.

Nonetheless, the distances cannot always be known. Unless this is possible, there is an alternative way to calculate it. Rather than dispatching vehicles based on the shortest distance rule, the dispatching decision could be made based on the shortest travel time. The decision is to dispatch vehicle such that

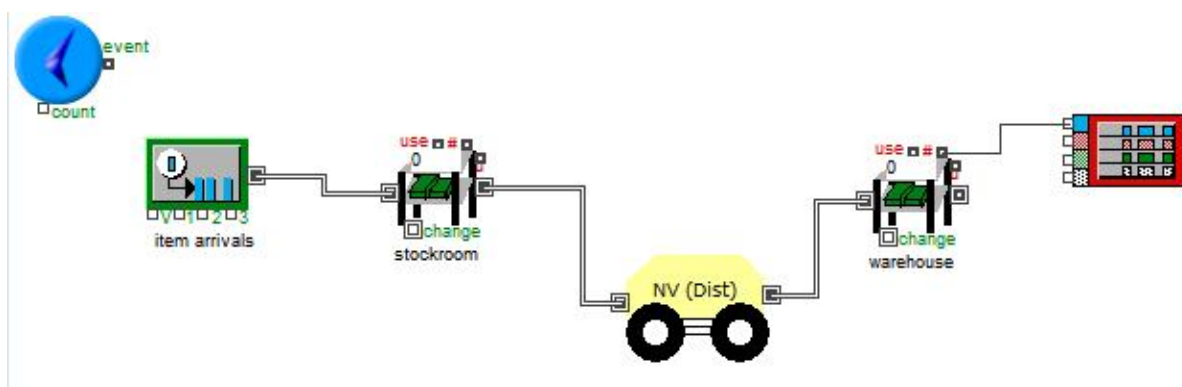
$$d_i = \min\{s_i/t\} \text{ for all idle vehicles } i,$$

where s is the travelling speed of vehicle i .

At the broadest level of detail, Davis (1986) assumed that AGVs have a constant speed for all movements within the system. In practice, many vehicles travel at a faster rate when unloaded than when carrying a job. Also, some vehicles experience decreases in speed as the battery wears down. In all systems, for a vehicle to start, stop, or negotiate turns, there are distinct deceleration and acceleration phases. Depending upon the decisions which are to be made based on the simulation analysis, as well as the actual significance of these factors in overall performance, it may not be necessary to include these details in the model.

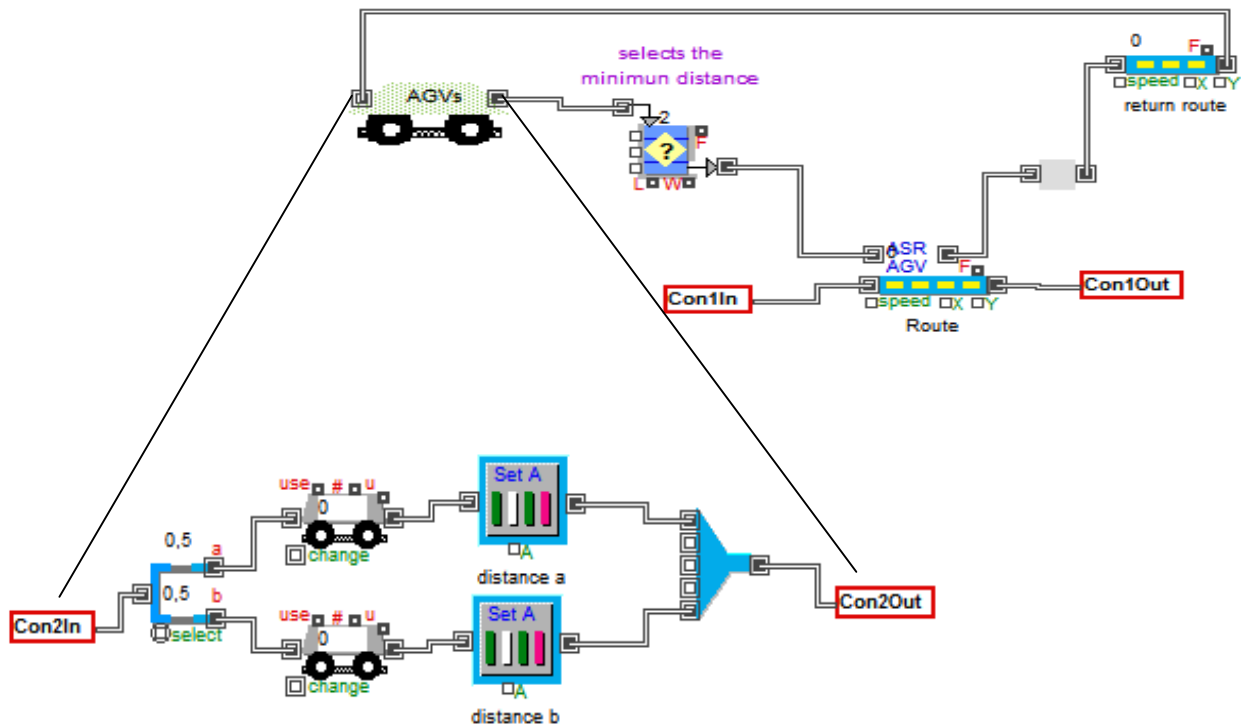
Consequently, provided that the vehicle speed is known and combined with the current time of each vehicle, it is easy to estimate which one is placed the nearest.

i. About the NV (distance) model



Picture 1f - The NV (distance) model

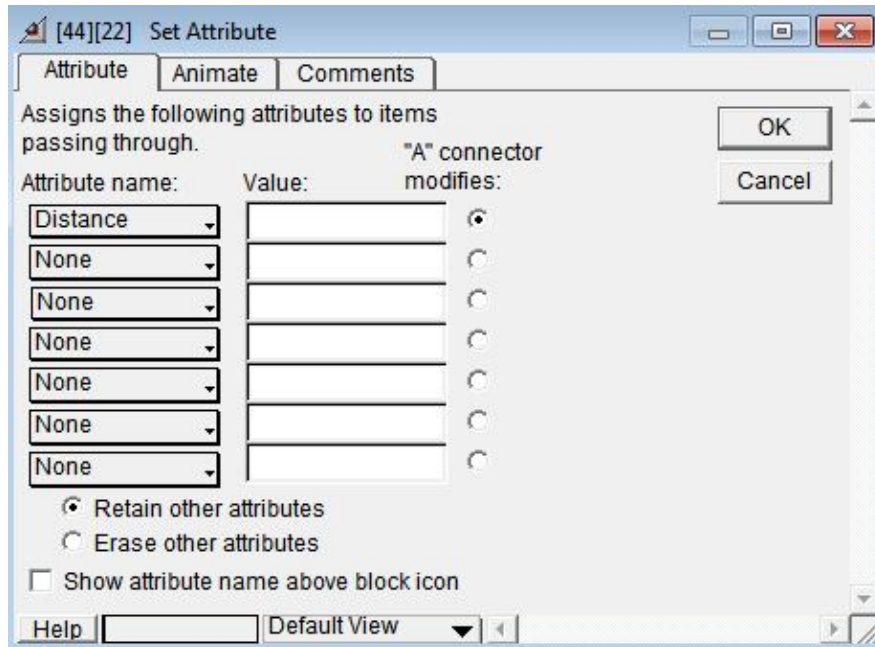
Nearest Vehicle (distance) model simulates the cups labeling operation by selecting the closest vehicle to the demand point. Specifically, picture 1f shows the model which consists of the blocks that were also met to the former models. Thus, the unlabeled cups begin their route from the generator block and end at the "NV (distance)" hierarchical block which is presented in picture 2f.



Picture 2f - The "Nearest Vehicle (distance)" hierarchical block

As it is shown, the cup enters the block from the "Con1In" connector and waits for a vehicle. At the same time, the vehicles start to move. Their travel begins from the hierarchical block "AGVs". This block, represents the different types of the AGVs which exist and run in the system. Different vehicle types arise from the fact that each vehicle starts its route from vehicle generators, which are located in different place from each other. The "AGV" blocks, in the hierarchical one, will be as many as the existing different locations in the system. As far as the location concerned, there is no specific attribute that can be declared in the generator block of the vehicles. This is the reason why there is the "Set Attribute" block right after every "AGV" block. Subsequently, this block sets the distance of the vehicles' initial position (distance a, distance b, etc), depending on their location from the queue, which is basically the demand point.

The next picture is the "Set Attribute" block's dialog. This dialog enables the vehicle to be assigned with many attributes apart from the ones that can be declared in the "AGV" block. Consequently, the vehicle is assigned with the desirable attribute, which is the distance from the demand point.

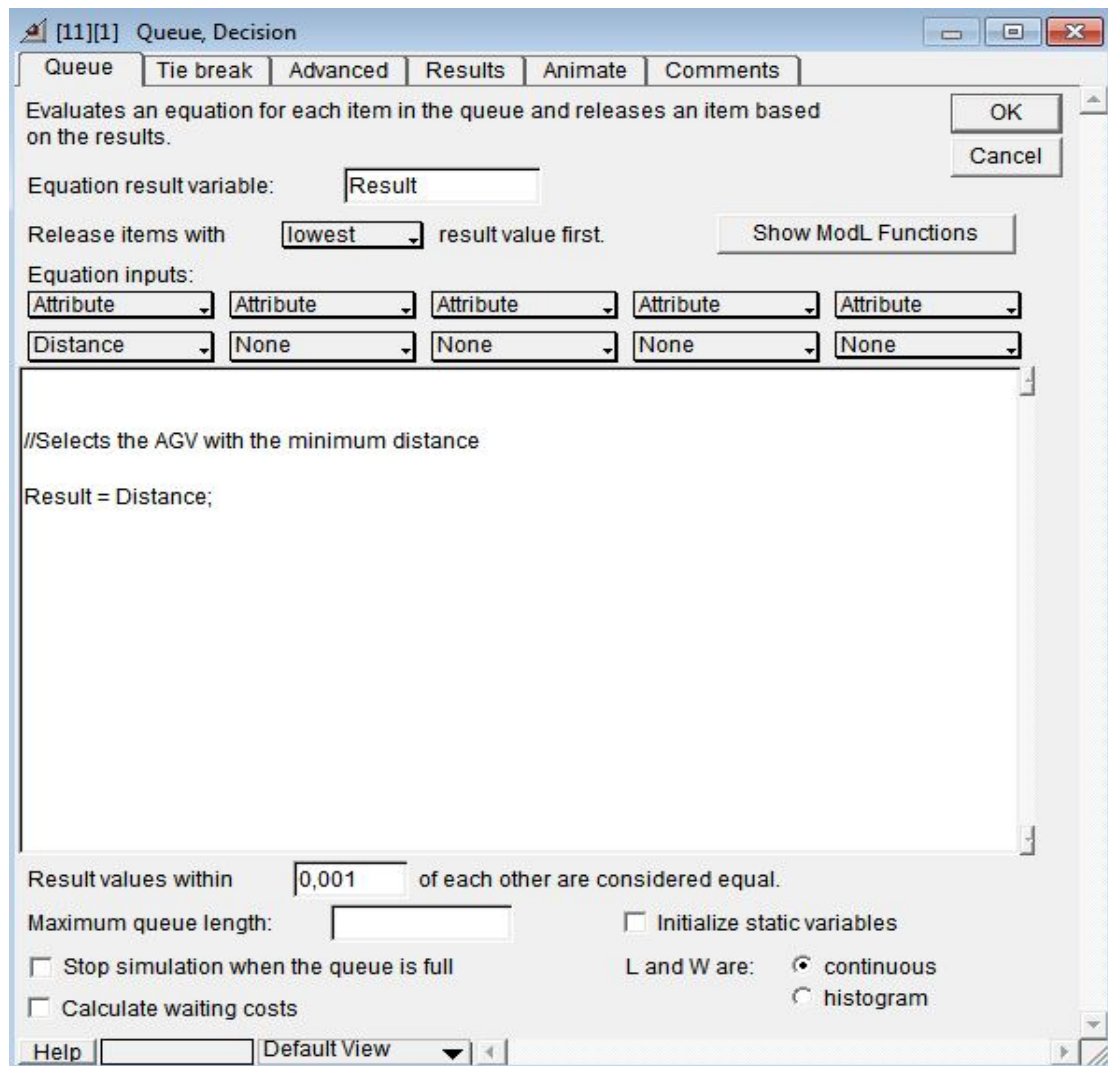


Picture 3f – **The "Set Attribute" block**

After the vehicles exit the "AGVs" block, they enter a queue block. At this point of the system, all the vehicles' distances from the demand point are collected and the AGV with the minimum one is dispatched. Apparently, this fact can be proved from picture 4f. This dialog enables the system to gather the different distances, compare them and select the vehicle with the minimum one. Therefore, there is no particular equation that must be calculated as the one and only declared attribute is distance.

Assuming that the closest vehicle is dispatched, it receives the waiting item, transfer it to carry out the assigned job and finally release it. As soon as the vehicle releases the item, it returns to its initial position.

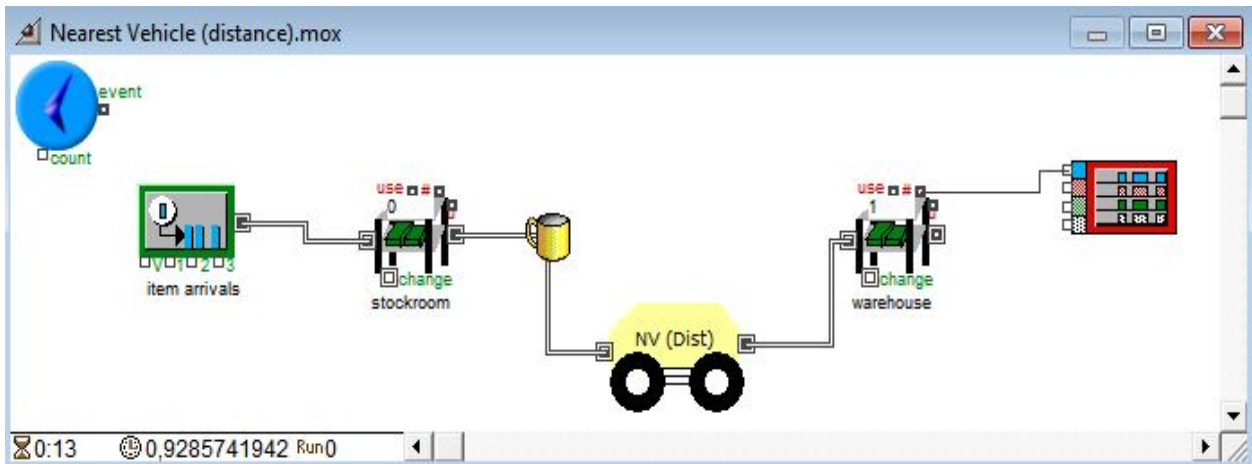
Finally, in order to avoid all the vehicles to return to the same "AGV" block, it is placed a "Select Output" block that gives the same probability for both the top and the bottom output.



Picture 4f – **The "Queue Decision" block**

ii. Running the NV (distance) model

In this case, it is assumed that there are 200 unlabeled cups that have to be labeled, 20 AGVs and 200 time units available. Below, it is shown that there is a cup travelling to the "NV (distance)" block.



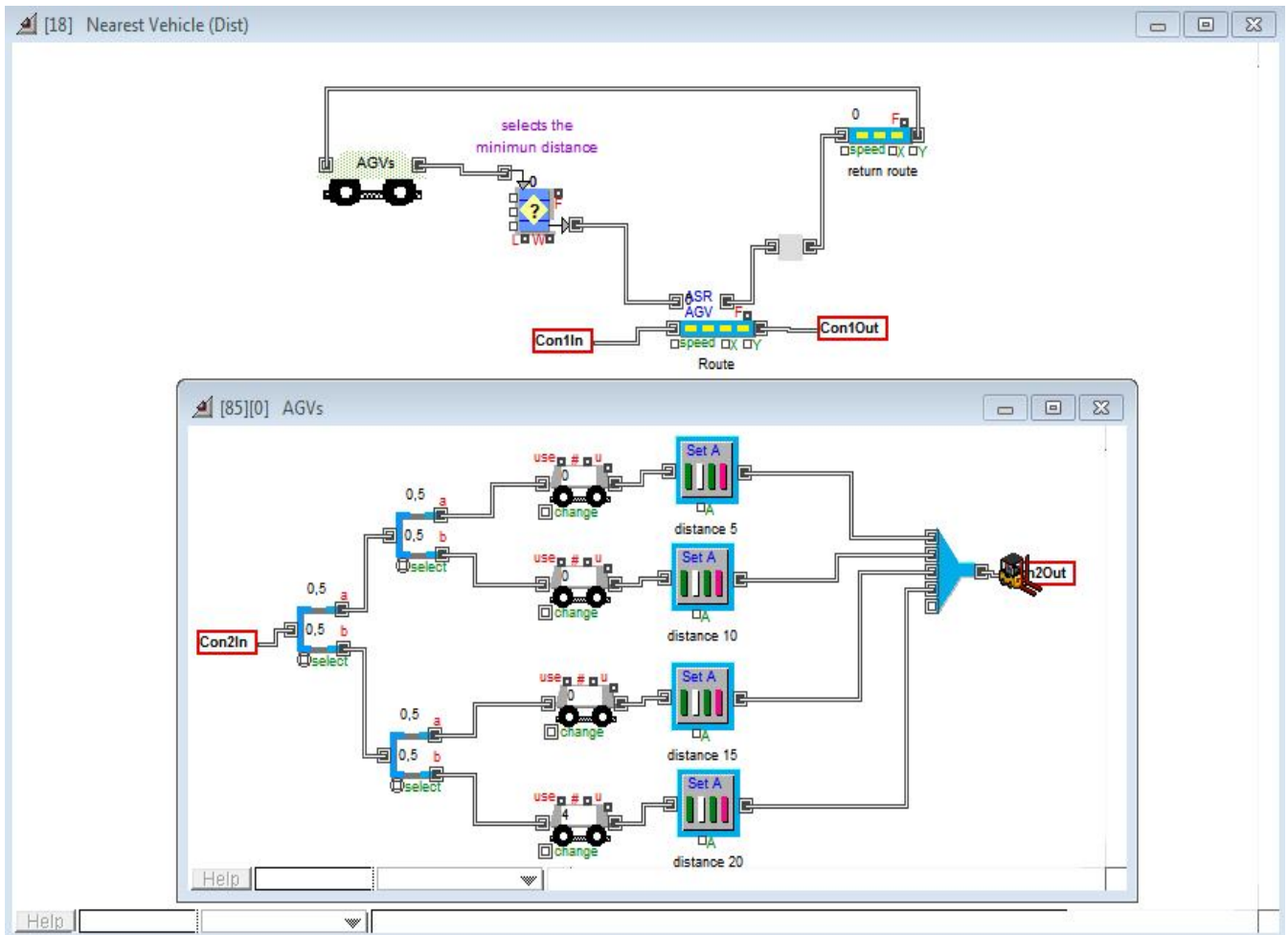
Picture 5f - The NV (distance) model

What is more, a vehicle has started its travel and is going to receive a cup. As soon as it receives it, the AGV drives the last one to the labeling machine. Afterwards, the labeled cup is led to the exit of their route, in order to be transferred to the warehouse.

Picture 6f depicts the entire procedure that takes place in the hierarchical block. To begin with, it has been assumed that there are 20 AGVs. Each vehicle block generates 5 AGVs. Since every AGV starts its travel, it passes through the "Set Attribute" block which assigns the vehicle with the distance attribute. In the particular simulation run they are set 4 blocks of AGVs which are located 5, 10, 15 and 20 distance units from the demand point. Thus, vehicles pass through, are assigned with the distance attribute and end up in the queue.

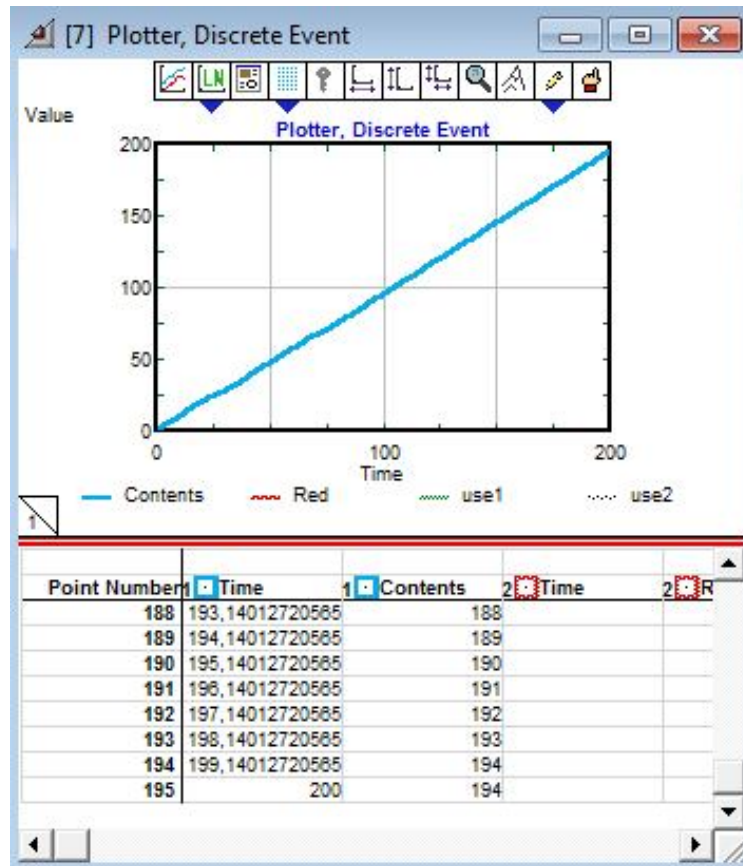
As it obvious from the picture below, it is plausible for the "AGV" blocks to be any number that is needed for the system. However, no matter how many vehicles there are in the system, it is important to have the same possibilities to enter at any "AGV" block at their return.

Furthermore, the referred queue is the place where the algorithm of the NV (distance) model selects the appropriate vehicle. After the selection the AGV is directed to the "Route" block which is where the unlabeled cup waits. At the moment it arrives there, it receives the cup and transfers it to the labeling machine. When the job is done, the AGV releases the labeled cup, and travels to its initial position.



Picture 6f - **The NV (distance) model**

When the time is 199,14012 units, there are 193 labeled cups in the warehouse ready for sale (Picture 7f), whereas the 194th labeled cup will get in the warehouse when the time will be about 200 units.



Picture 7f - **The simulation plotter**

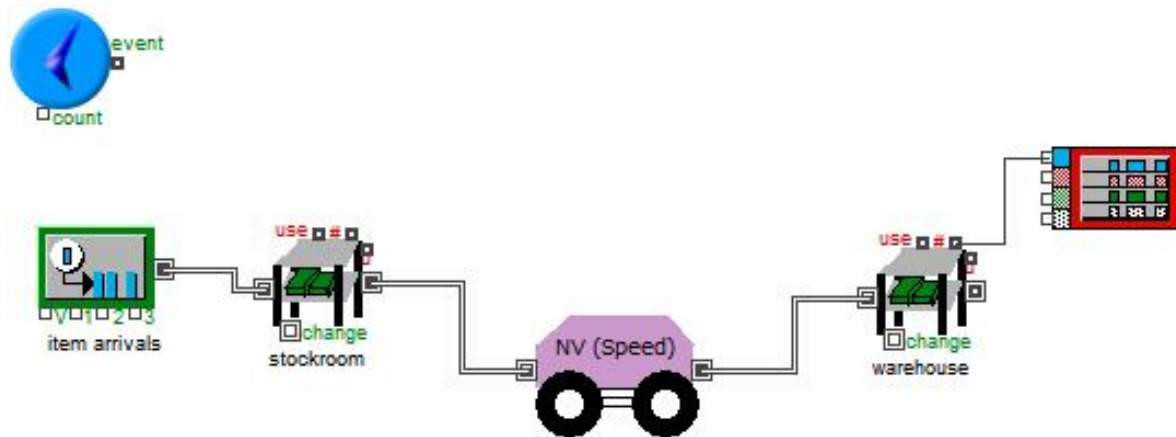
iii. About the NV (speed) model

Nearest vehicle can be localized with a different way, as well. Unless the exact position of each vehicle is known, it can be calculated though. This calculation can be depended on a fundamental law of physics. This law demonstrates that the distance is the product of speed and time. This is written in mathematics:

$$u_i = s_i \times t, \text{ for every } i \text{ vehicle,}$$

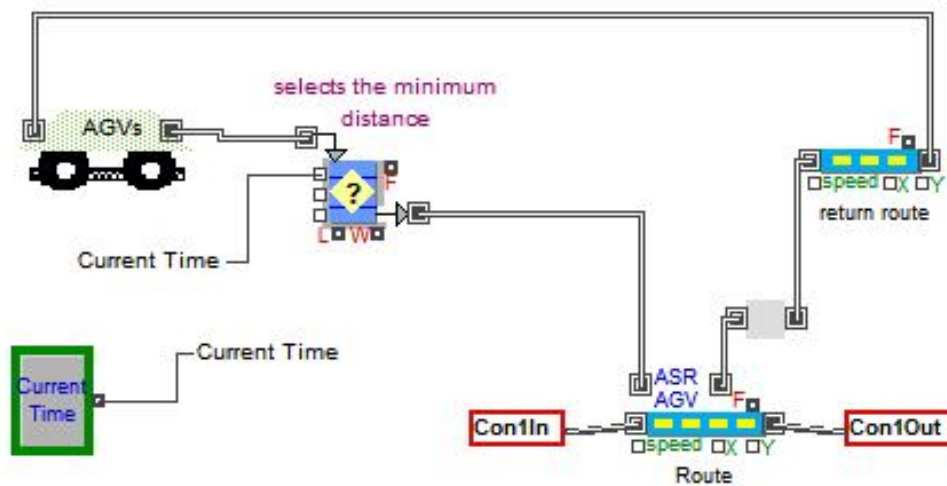
where u_i is the speed of the vehicle, s_i is the traveled distance and t the time that took it to travel the s_i distance with speed equal to u_i .

Consequently, it is quite important to explore the NV (speed) model. At first sight, as it is shown in picture 9f, the model consists of the executive block, the generator, two stock blocks, a hierarchical one and a plotter.



Picture 9f - **The NV (speed) model**

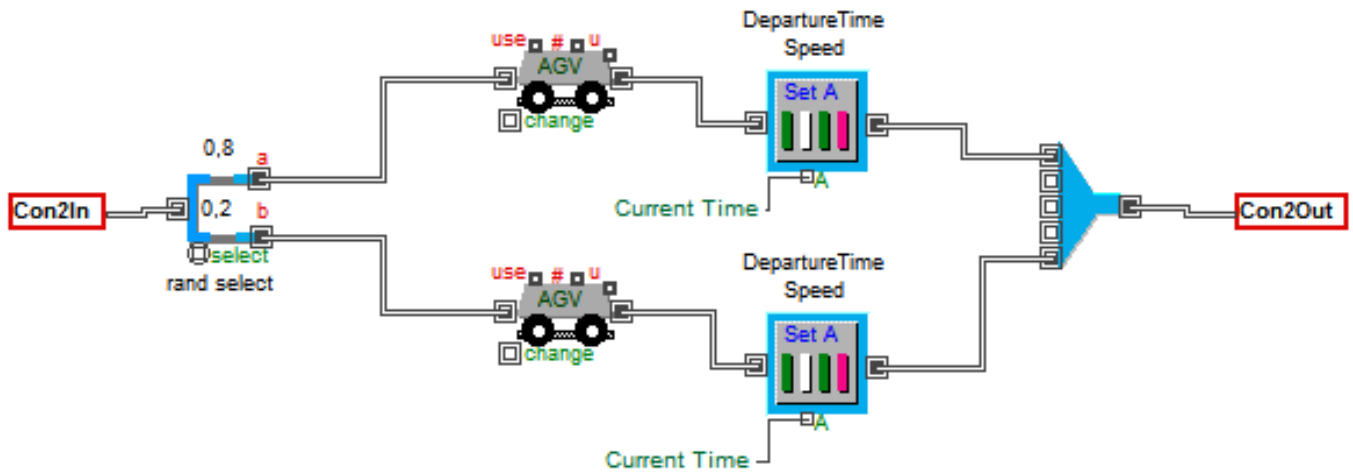
As soon as the items arrive at the hierarchical block, they enter through the "Con1In" connector, as picture 10f shows. At the same time a vehicle is moving towards "Route" block to receive it. The AGV receives the item and set it free at the end of the "Route" block in order to go to the warehouse, while the AGV returns from a different route to the hierarchical block.



Picture 10f - **The "NV (speed)" hierarchical block**

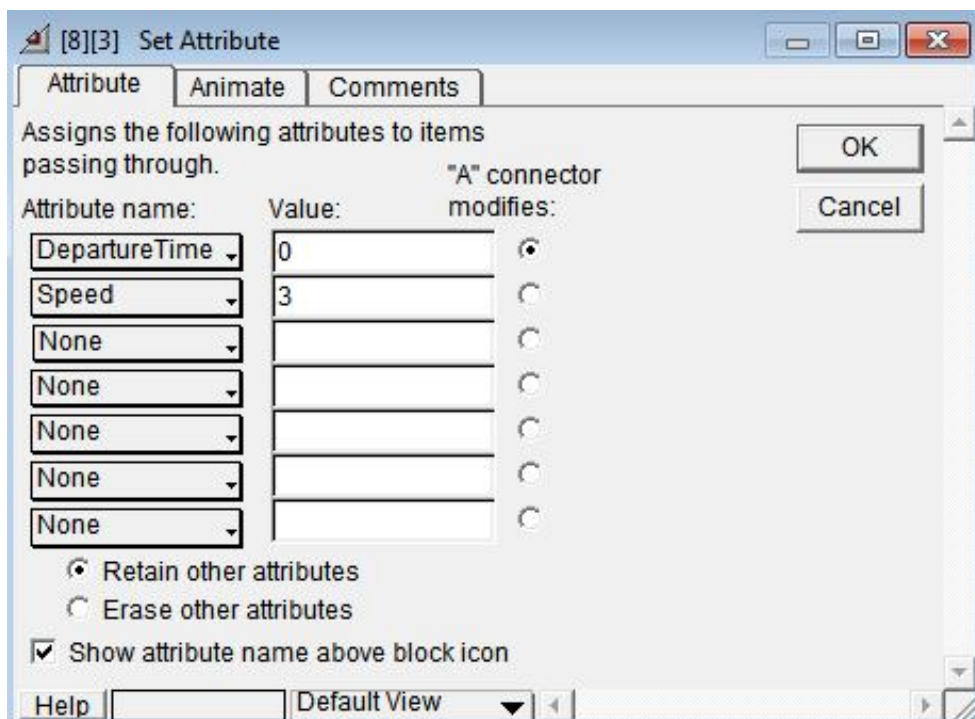
In addition to this procedure, it is remarkable to explicate what exactly happens in the hierarchical block "AGVs" combined with the "Queue Decision" block and the right vehicle is finally dispatched.

At first it is important to explore the "AGVs" hierarchical block. Inside this block, there are "AGV" blocks which are connected with a "Set Attribute" block.



Picture 11f - The "AGVs" hierarchical block

This block assigns with extra attributes the AGVs (speed, departure time). The relevant dialog is given by the picture below.

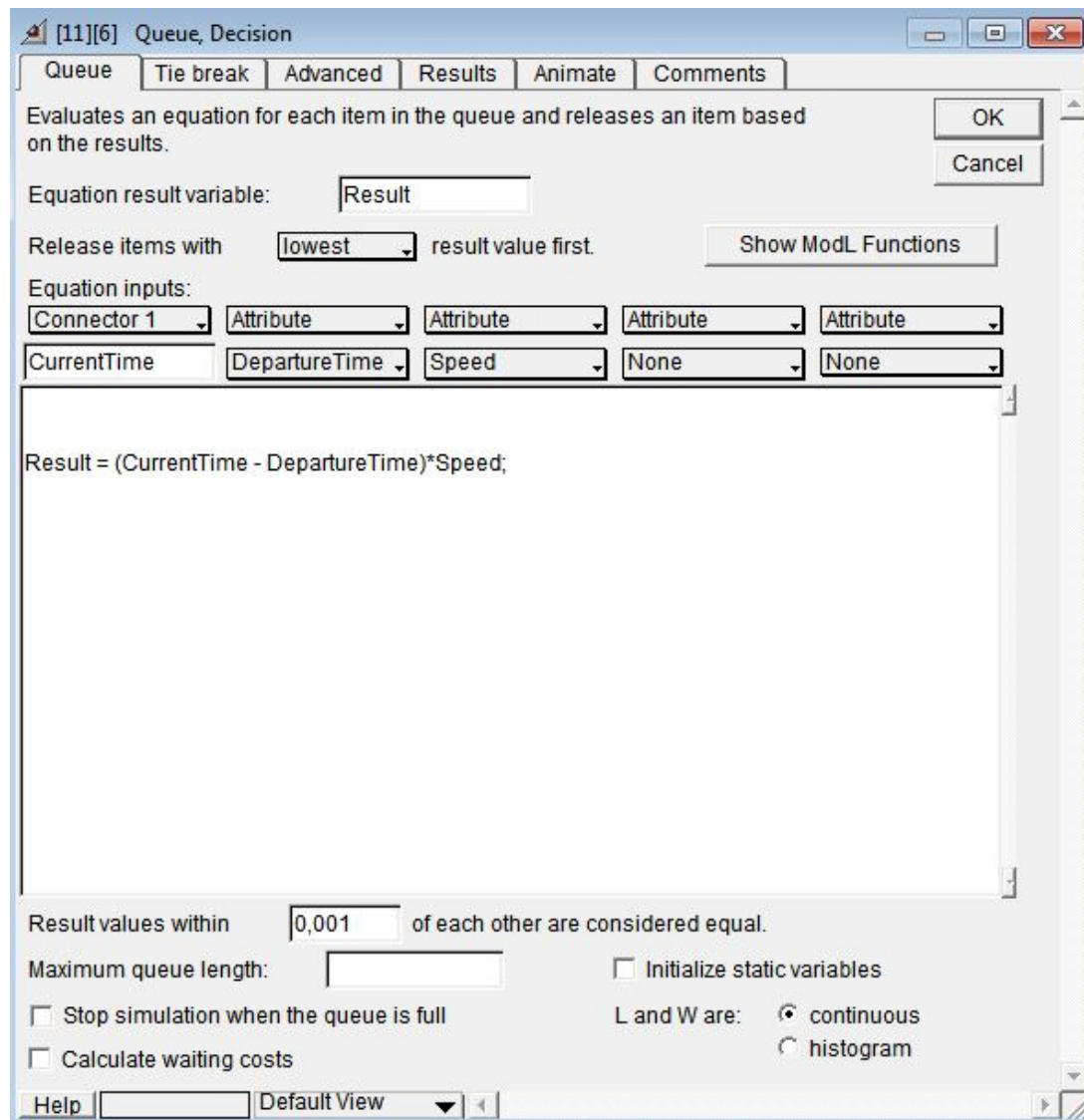


Picture 12f - The "Set Attribute" block

As the open dialog shows, the AGVs are assigned with two attributes. The first one is Departure Time that has not a fixed value. Instead, its value is formed according to the value of the "System variable" block (picture 12f). Specifically, this connection represents the vehicle's time stamp that has at the beginning of its travel. What is more, the second attribute, speed, is a constant number.

Consequently, when the vehicle passes through the "Set Attribute" block, it is assigned with two new attributes, a time stamp which denotes its departure time and a value for its speed. Under the assumption that the vehicle speed is constant during the simulation, then it is quite plain to calculate the minimum distance.

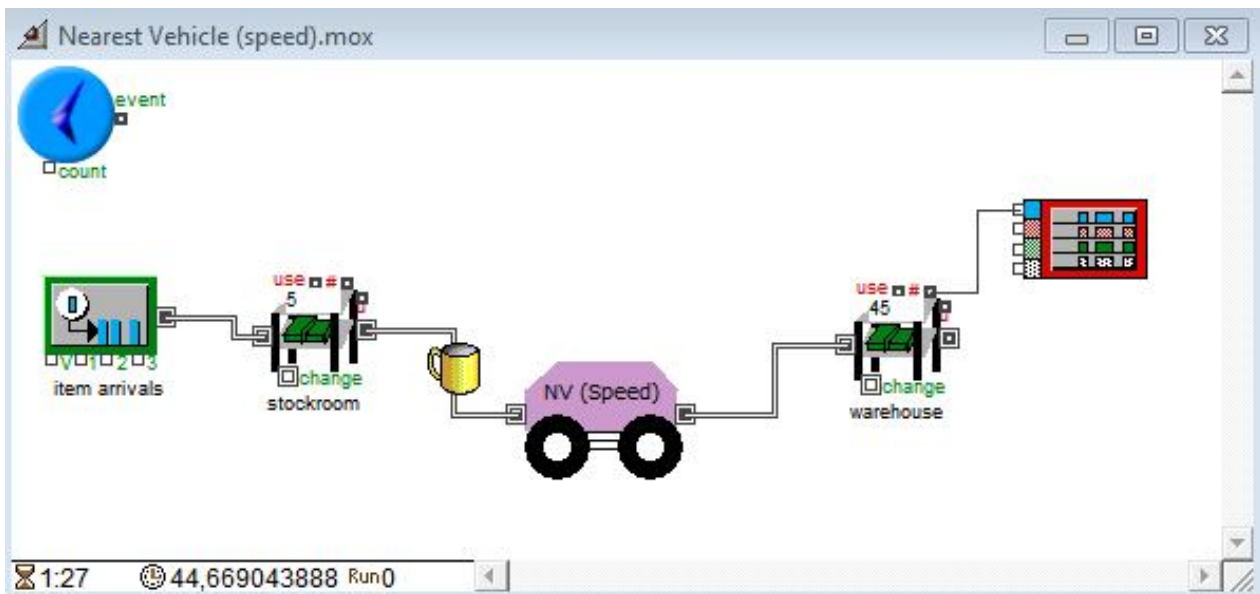
Afterwards, the vehicles enter the "Queue Decision" block, where the AGV with minimum distance is selected. This is achieved with the aid of the equation in the queue tab, as the following picture shows. The equation has three inputs; speed, departure time and current time. According to the fundamental law of physics, distance is calculated when the speed vehicle is multiplied with its travel time. Travel time accrues from the abstraction of the Departure time from the Current Time. Departure Time and speed are attributes that has the vehicle in it when it passes through the specific block. As far as Current Time concerned, it takes its values from the connector1, which is connected with the System Variable "Current Time" as picture 10f denotes. Finally, since the result is calculated, the AGV with the lowest one is chosen to be dispatched.



Picture 13f - The "Queue Decision" block

iv. Running the NV (speed) model

It is assumed that there are 200 unlabeled cups, 15 AGVs and 200 time units. When the time that has elapsed is 44,66904 units, an unlabeled cup is approaching the hierarchical block, in order to be received by an AGV and get labeled. So, it enters the "NV (speed)" block through the input connector and stops at the "Route" block, where it waits for a vehicle.

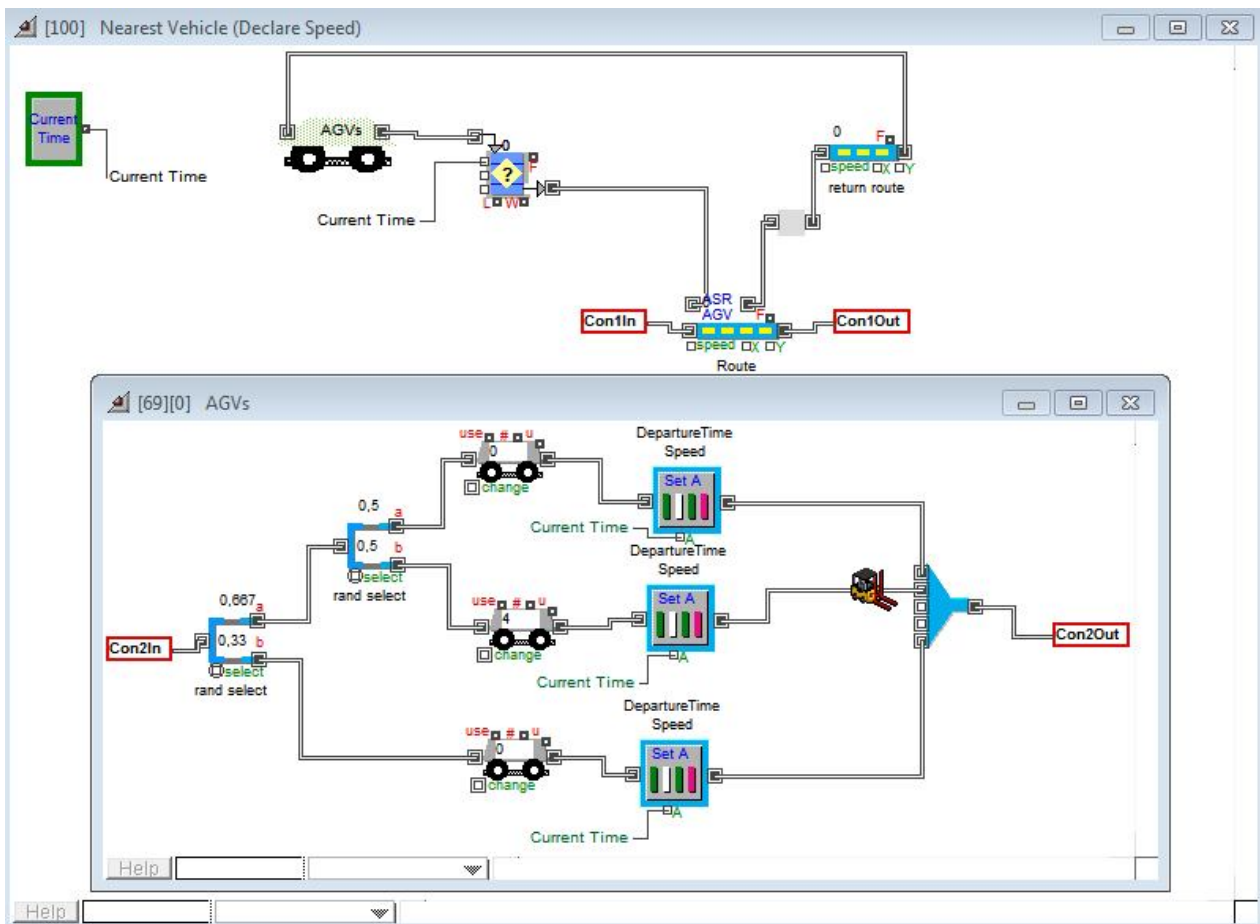


Picture 14f - The NV (speed) model

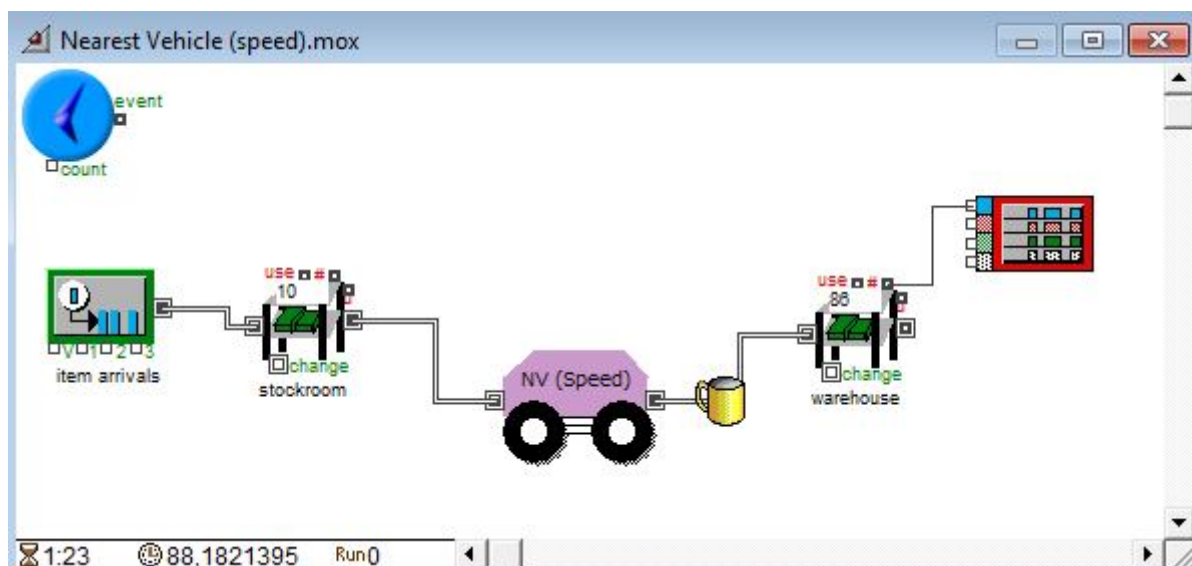
As soon as a job occurs and a vehicle is demanded, AGVs start to move towards the demand point (picture 15f). Since AGV are generated, they pass through the next block in the raw, where they are assigned with two new attributes, totally necessary for the model; speed and departure time. Beginning from the vehicle on top, vehicles' speeds are set 3, 4 and 5 m/s respectively. The departure times are formed depending on the time that the AGV started its route. The described procedure takes place in the "AGVs" hierarchical block. Thereafter, the vehicles exit and are gathered in the queue, where they wait to be dispatched.

Nevertheless, in order to work properly, it is not enough for the algorithm the vehicles' speed and departure time. This is the reason why "Queue Decision" block uses the values that are formed by the block's connector, which is "Current Time". Thus, the algorithm calculates the travel time of the vehicles (by subtracting departure time from the arrival time in the queue – current time) and multiplies it with their speed. The AGV with the minimum result is the one that will be released and receive the cup.

Consequently, it receives the cup, transfers it to the labeling machine and set it free to go to the warehouse (picture 16f). This picture also shows that when the time of the simulation is 88,18213 units there are 10 unlabeled cups in the stockroom and 86 labeled cups in the warehouse already.

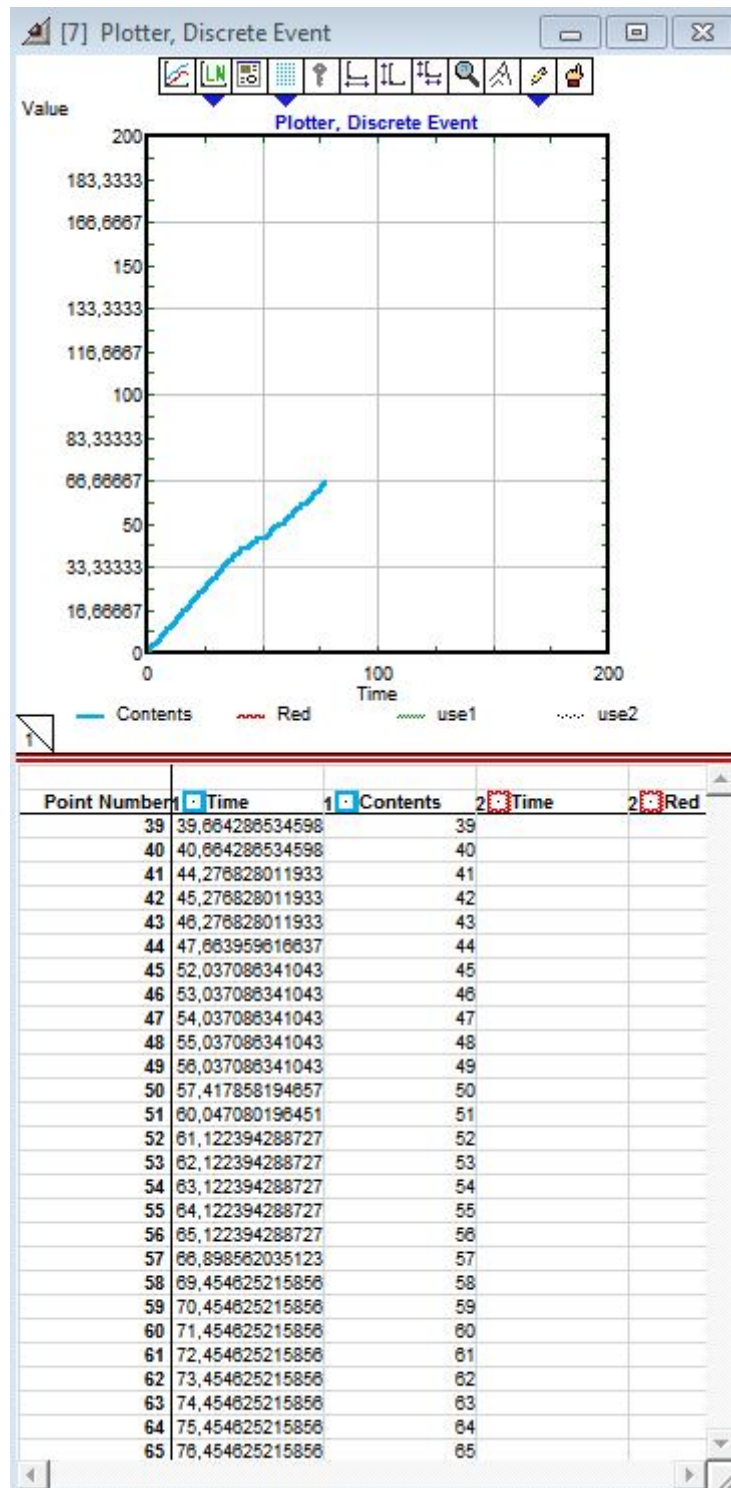


Picture 15f - The "NV (speed)" hierarchical block



Picture 16f - The NV (speed) model

Furthermore, the simulation plotter (picture 17f) presents the times that each labeled cup entered the warehouse in detail. For example, here, it is depicted that when the simulation time was 76,454625 units the 65th labeled cup entered the last simulation block.



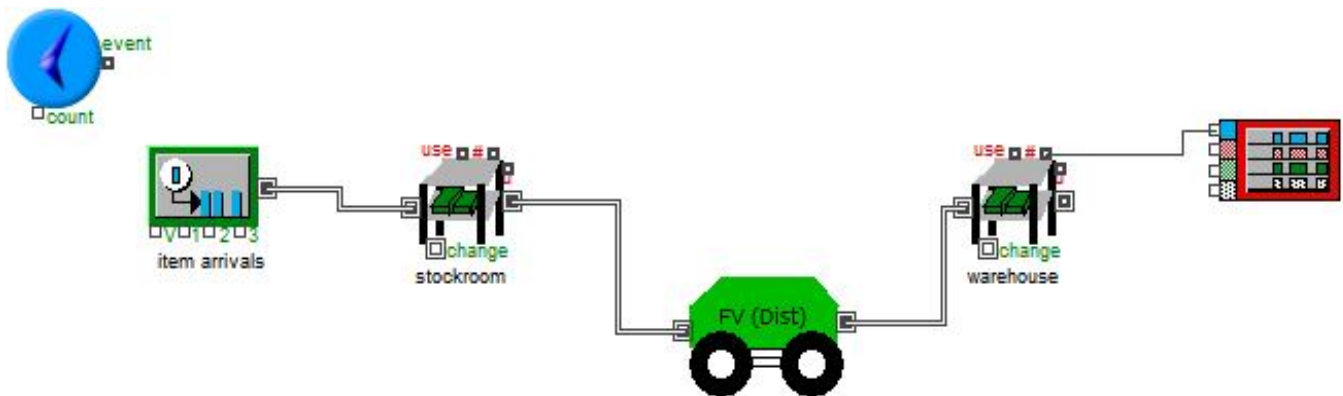
Picture 17f - The simulation plotter

g. Farthest Vehicle (FV)

This model is the exact opposite to the previous one. Farthest Vehicle model can be divided in two categories, as well: the first one considers that the position of every vehicle in the system is known, whereas the second one ignores the vehicle's position, yet calculates it with the aid of the vehicle speed, assuming that it is steady during a simulation run.

i. About the FV (distance) model

Analyzing the behavior of the model is essential to mention that its algorithm selects to dispatch a vehicle that lies the farthest from the demand point.

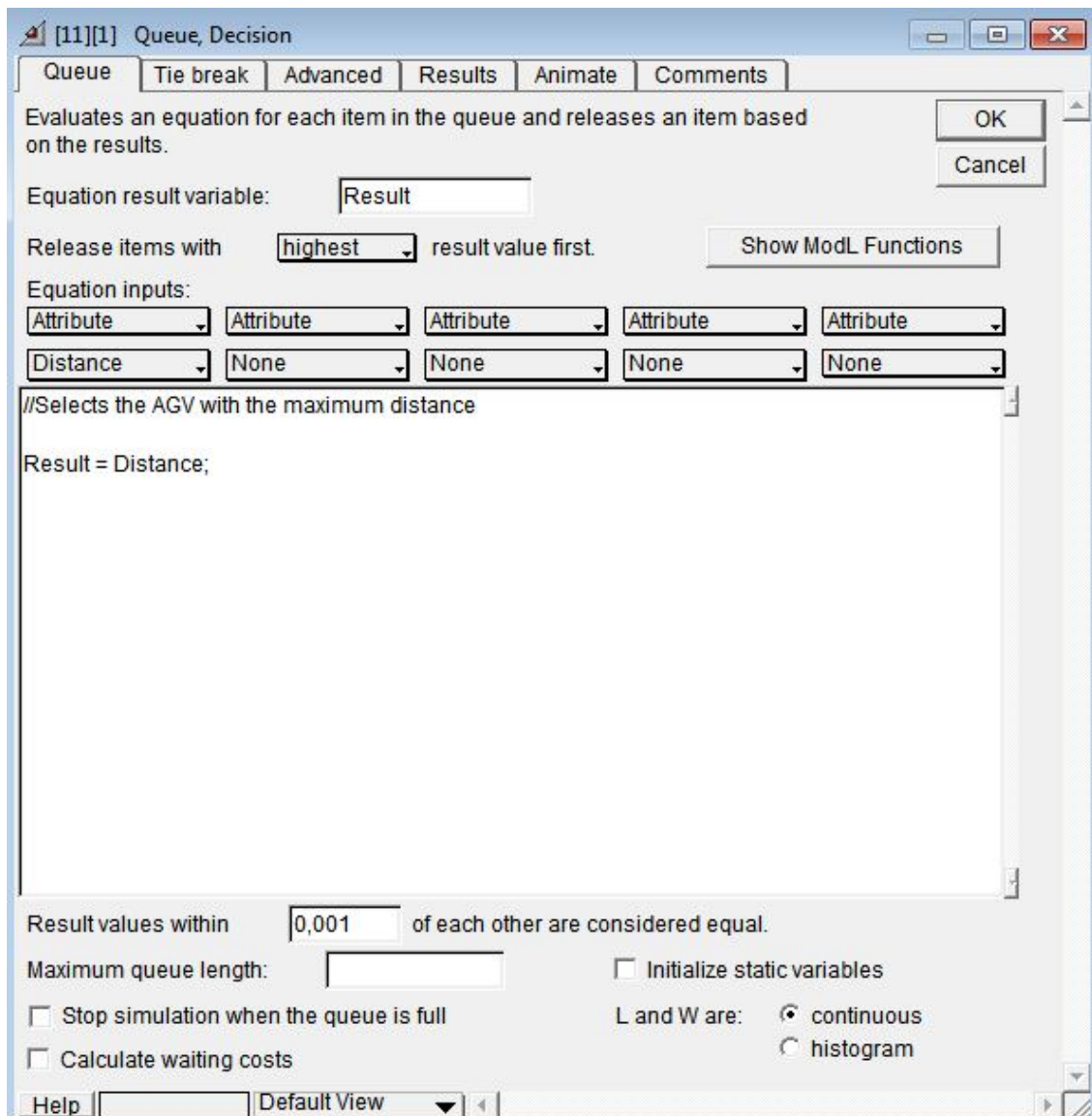


Picture 1g - The FV (distance) model

Consequently, there are the blocks that simulate the basic movement of the items and there is the hierarchical block that contains the algorithm of the model and selects the appropriate vehicle every time. The items start to move from the generator block. After a while they meet the stock block, where they are stocked until they enter the FV (Dist) hierarchical block. As soon as they leave this last block, they are stocked once again in the stock block waiting to be sold.

Nevertheless, the activity in the hierarchical block determines whether or not the algorithm will work. Picture 2g is the inner depiction of the hierarchical block. When the items move, they stop at the front of the "Route" block waiting for a vehicle to deliver them.

On the other hand, AGVs begin their route from the "AGVs" block which is hierarchical, too. In this block there is one, two or more "AGV" blocks, each of which



Picture 3g - **The "Queue Decision" block**

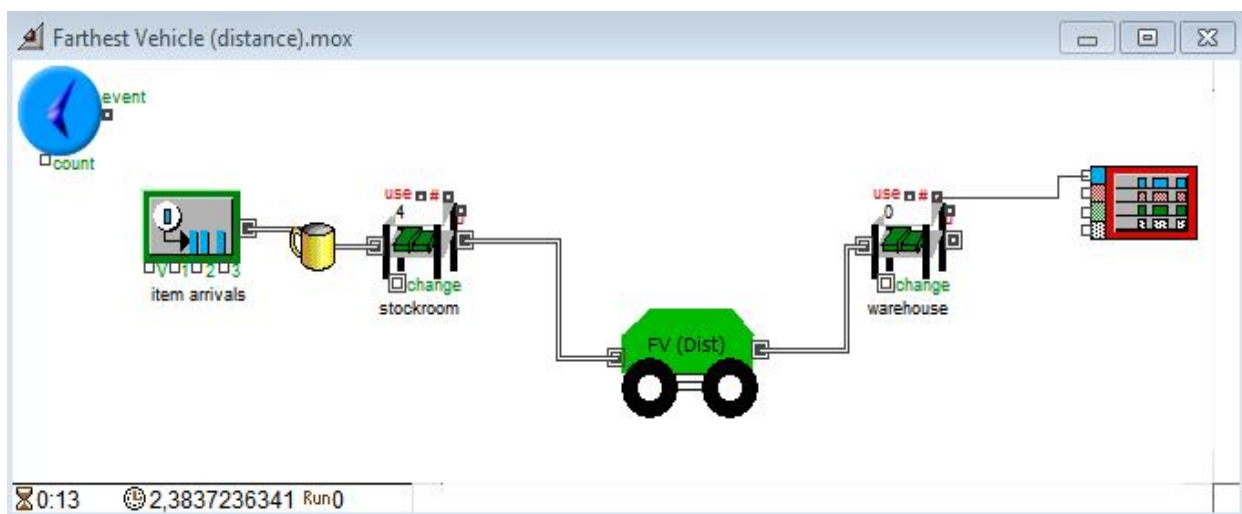
Provided that the farthest vehicle is dispatched, it receives the item at the "Route" block and after the completion of the job, it releases it to go to the warehouse. The AGV is now empty and ready to return to its initial position.

At last, when the vehicle is at the "Con2In" connector meets a "Decision Output" block. This block provides equal possibilities for the vehicle to return either to the top or to the bottom initial block.

ii. Running the FV (distance) model

It is assumed that there are 200 cups that have to be labeled and 20 AGVs that are assigned to load the unlabeled cups and transfer them to the labeling machine. Furthermore, there are 200 time units available for the whole procedure to be completed.

Picture 4g shows an unlabeled cup entering the stockroom when the time is 2.38372 units, yet there are already 4 cups in it.

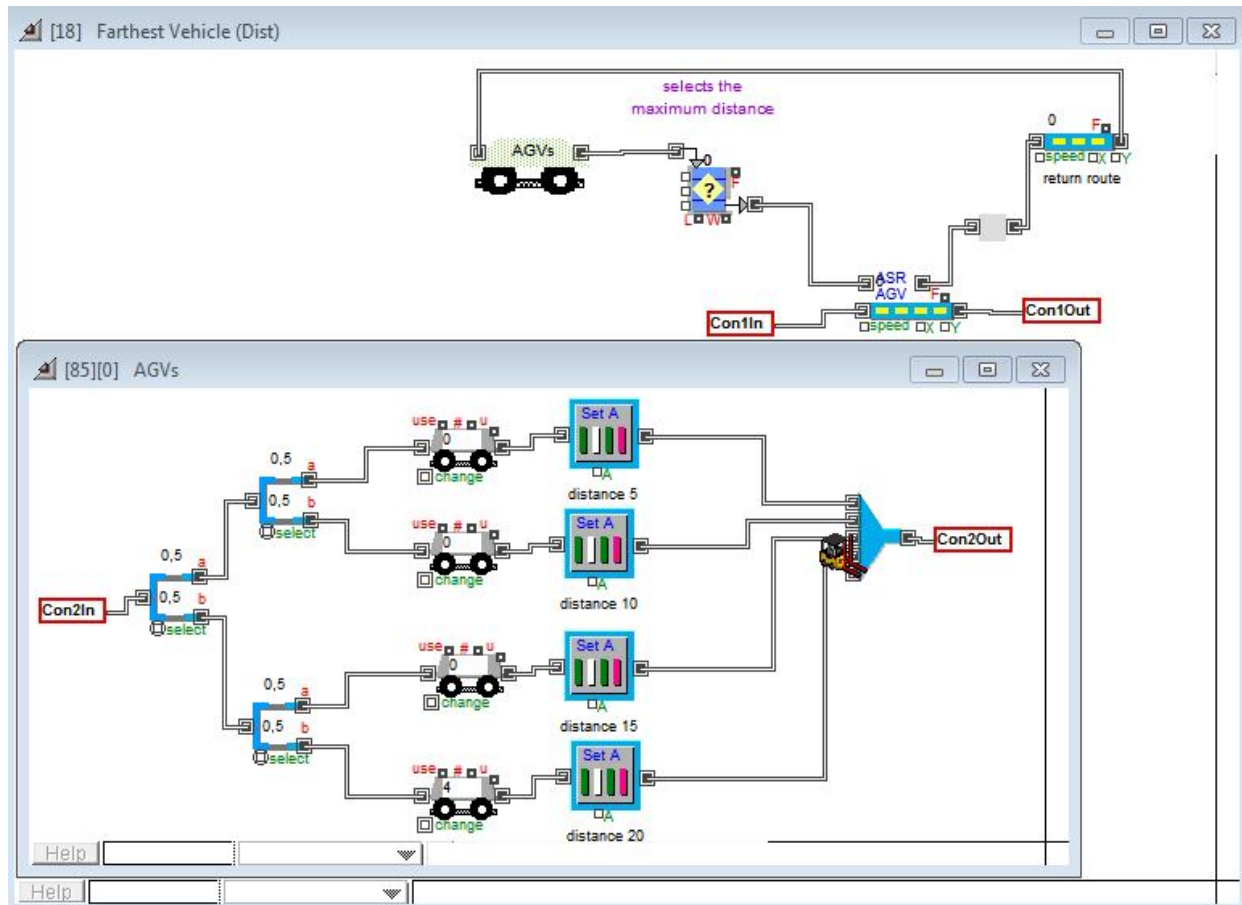


Picture 4g - The FV (distance) model

When the unlabeled cup enters the "FV (dist)" block, it stops in front of the "Route" block and waits for a vehicle. Picture 5g is the inner depiction of the hierarchical block. According to the assumptions for 20 AGVs, there are 4 "AGV" blocks each of which generate 5 vehicles with the same attributes. The only different attribute among the generated vehicles is their distance from the demand point. This is the reason why after every vehicle block there is a "Set Attribute" block which assigns the desired attribute (distance) to the block.

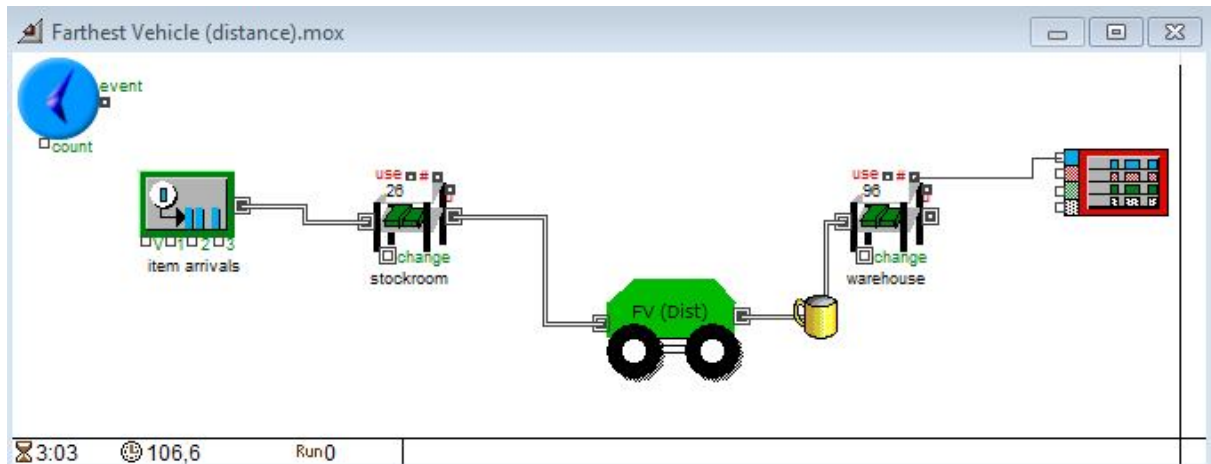
In the example, it seems that this moment a vehicle exits from the "AGVs" block coming from the bottom block which is placed 20 distance units from the demand point. However, there is more than one vehicle in the queue, waiting for the block to decide which of them will be dispatched. As soon as the farthest vehicle is dispatched, it is released and directed to the "Route" block to deliver the unlabeled vehicle.

After the completion of its assigned job (transferring the unlabeled cup to the labeling machine), it sets the labeled cup free to go to the warehouse and the vehicle follows its return route.



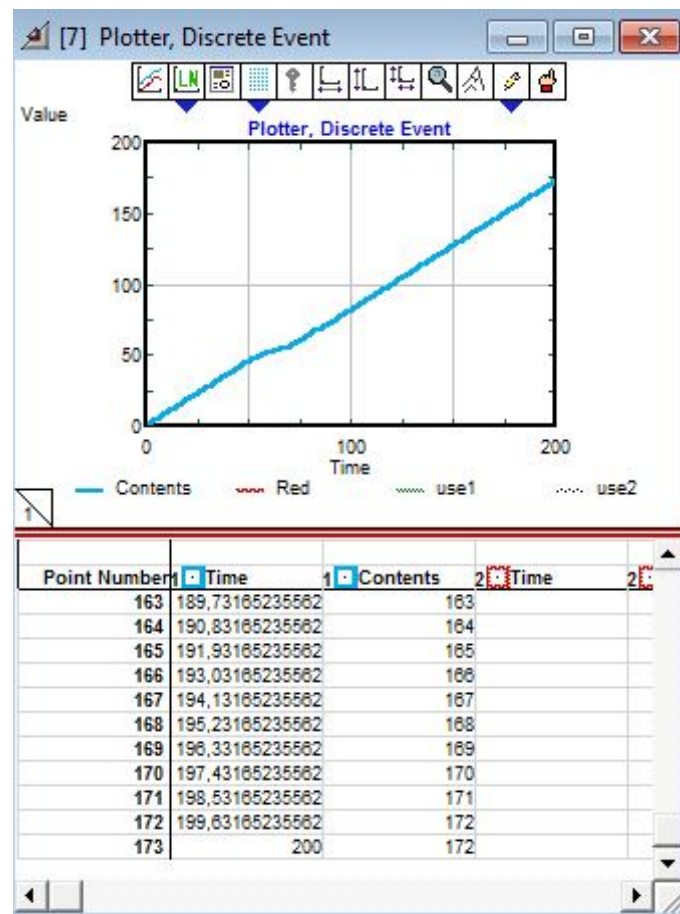
Picture 5g - The "FV (distance)" hierarchical block

When the time is 106,6 units, there are 96 labeled cup in the warehouse ready for sale, whereas there are 26 unlabeled cup waiting in the stockroom (Picture 6g).



Picture 6g - **The FV (distance) model**

In addition, the simulation plotter (picture 7g) presents the times that each labeled cup entered the warehouse in detail. For example, here, it is depicted that when the simulation time is 200 units the 172th labeled cup entered the last simulation block.



Picture 7g – **The simulation plotter**

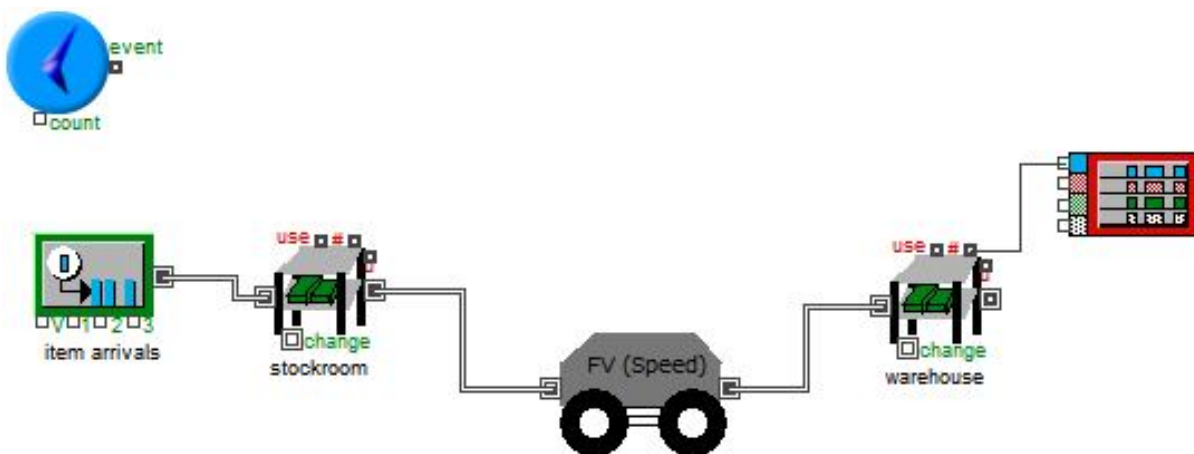
iii. About the FV (speed) model

Farthest vehicle can be localized with a different way, as well. Unless the exact position of each vehicle is known, it can be calculated though. This calculation can also be depended on the fundamental law of physics. This law demonstrates that the distance is the product of speed and time. This is written in mathematics:

$$u_i = s_i \times t, \text{ for every } i \text{ vehicle,}$$

where u_i is the speed of the vehicle, s_i is the traveled distance and t the time that took it to travel the s_i distance with speed equal to u_i .

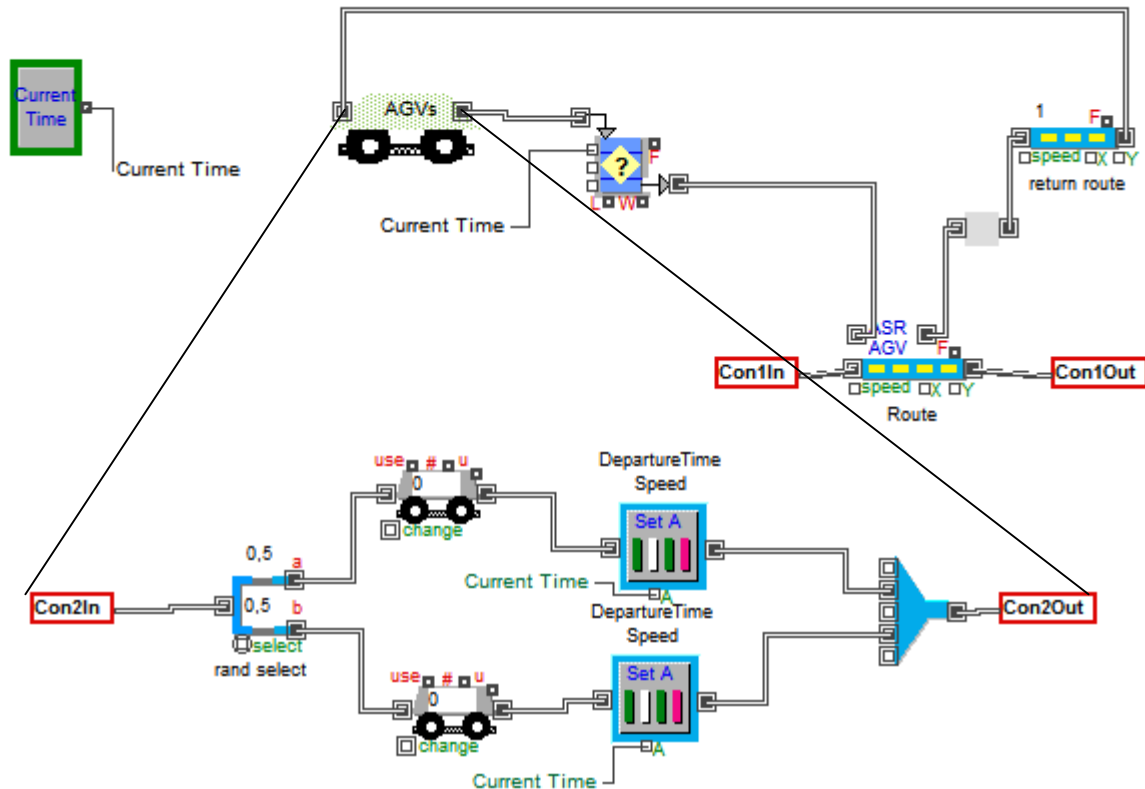
Picture 8g shows the main model which consists of the executive block, the item generator, two stock blocks, the simulation plotter and the hierarchical block "FV (speed)", the heart of the model.



Picture 8g - **The FV (distance) model**

As soon as the items arrive at the hierarchical block, they enter through the "Con1In" connector, as picture 9g shows. At the same time a vehicle is moving towards "Route" block to receive it. The AGV receives the item and set it free at the end of the "Route" block in order to go to the warehouse, while the AGV returns from a different route to the hierarchical block.

In addition to this procedure, it is remarkable to explicate what exactly happens in the hierarchical block "AGVs" combined with the "Queue Decision" block and the appropriate vehicle is finally dispatched.

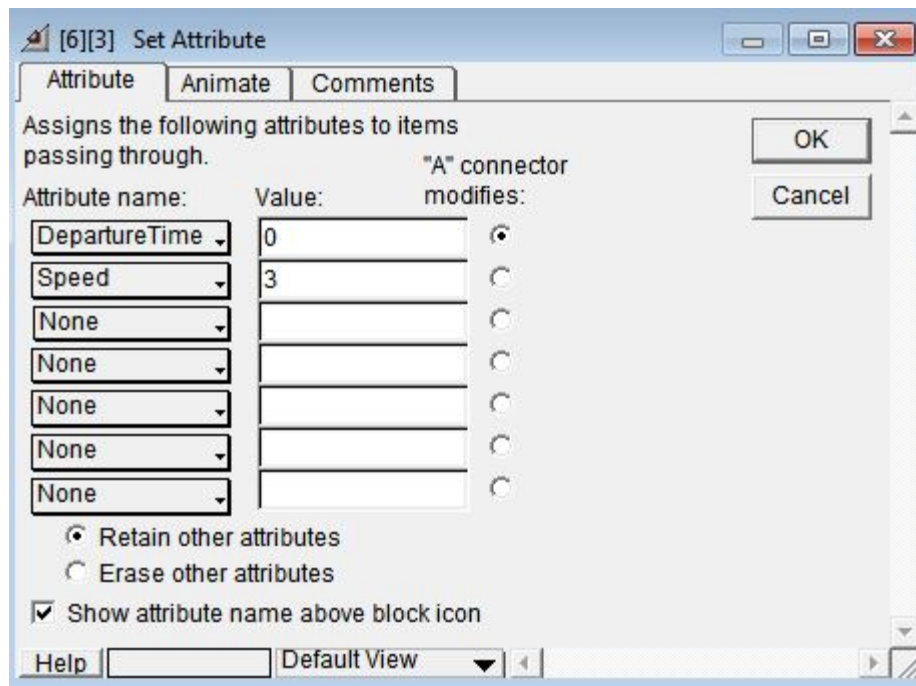


Picture 9g - The FV (speed)" hierarchical block

At first it is important to explore the "AGVs" hierarchical block. Inside this block, there are "AGV" blocks which are connected with a "Set Attribute" block. This block assigns with extra attributes the AGVs (speed, departure time).

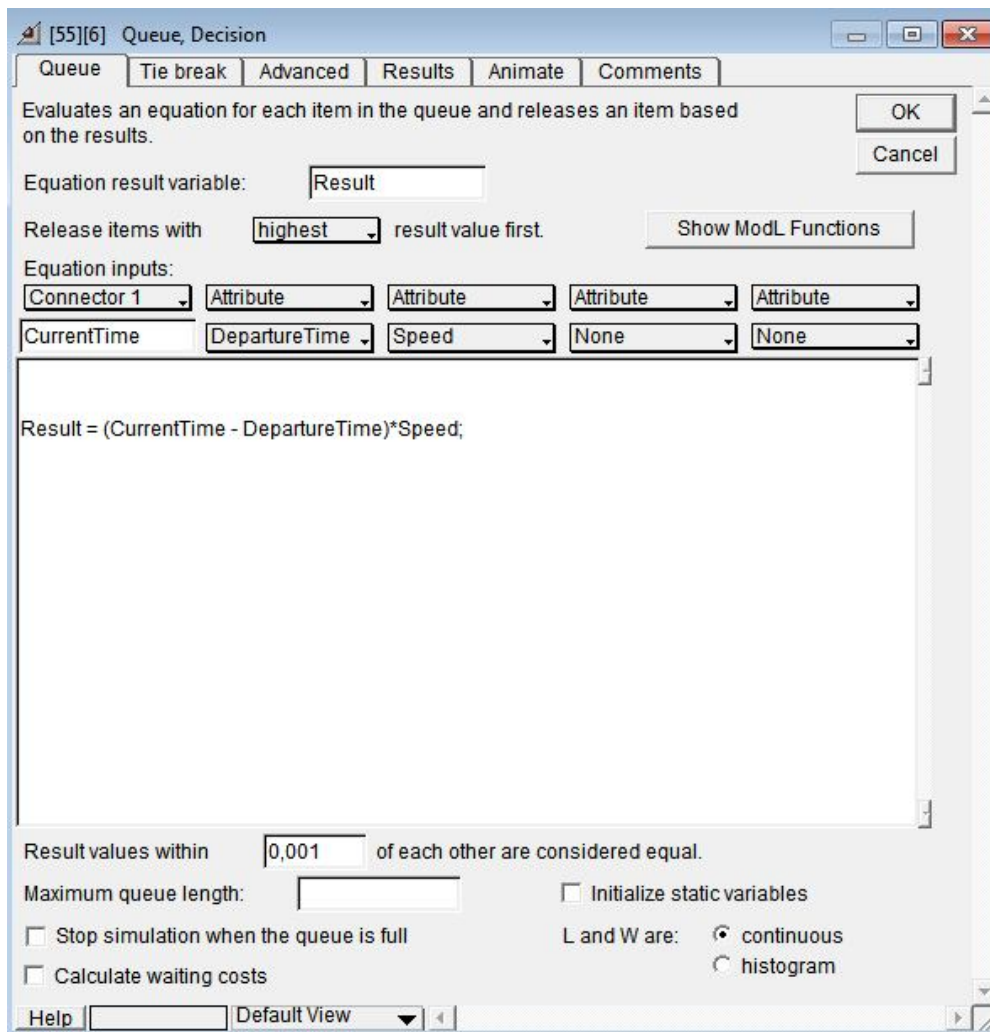
As the open dialog shows (Picture 10g), the AGVs are assigned with two attributes. The first one is Departure Time that has not a fixed value. Instead, its value is formed according to the value of the "System variable" block. Specifically, this connection represents the vehicle's time stamp that has at the beginning of its travel. What is more, the second attribute, speed, is a constant number.

Consequently, when the vehicle passes through the "Set Attribute" block, it is assigned with two new attributes, a time stamp which denotes its departure time and a value for its speed. Under the assumption that the vehicle speed is constant during the simulation, then it is quite plain to calculate the maximum distance.



Picture 10g - **The "Set Attribute" block**

Afterwards, the vehicles enter the "Queue Decision" block, where the AGV with maximum distance is selected. This is achieved with the aid of the equation in the queue tab, as the following picture shows.

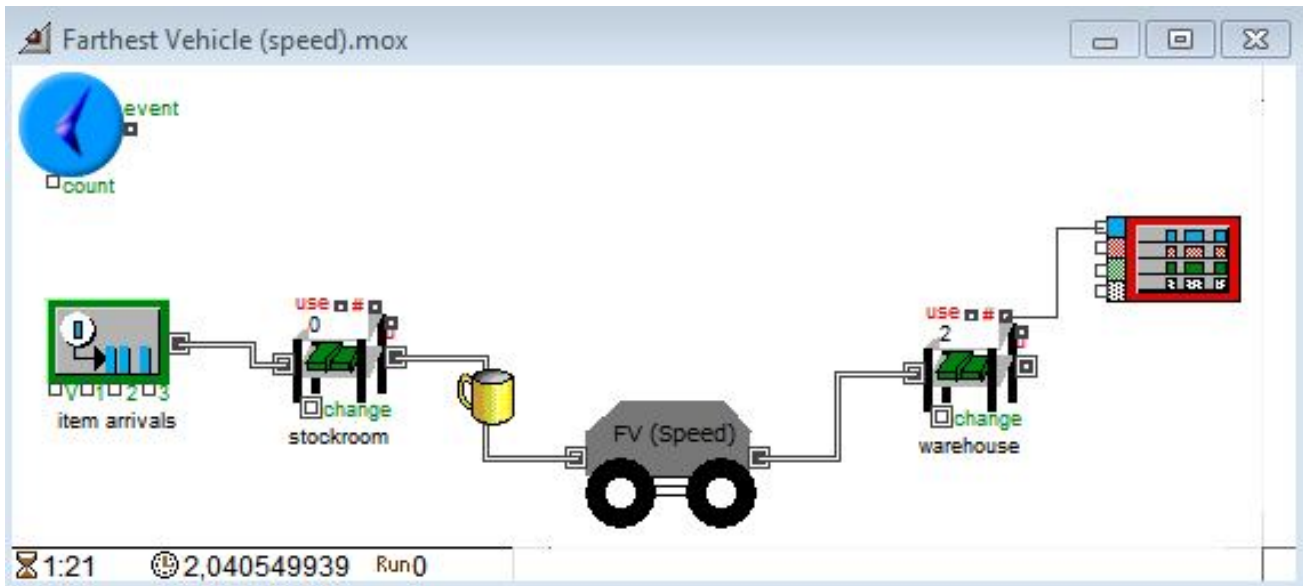


Picture 11g - The "Queue Decision" block

The equation has three inputs; speed, departure time and current time. According to the fundamental law of physics, distance is calculated when the speed vehicle is multiplied with its travel time. Travel time accrues from the abstraction of the Departure time from the Current Time. Departure Time and speed are attributes that has the vehicle in it when it passes through the specific block. As far as Current Time concerned, it takes its values from the connector1, which is connected with the System Variable "Current Time" as picture 9g denotes. Finally, since the result is calculated, the AGV with the highest one is chosen to be dispatched.

iv. Running the FV (speed) model

It is assumed that there are 200 unlabeled cups, 15 AGVs and 200 time units. When the time that has elapsed is 2,04054 units, an unlabeled cup is approaching the hierarchical block, in order to be received by an AGV and get labeled. So, it enters the "NV (speed)" block through the input connector and stops at the "Route" block, where it waits for a vehicle.



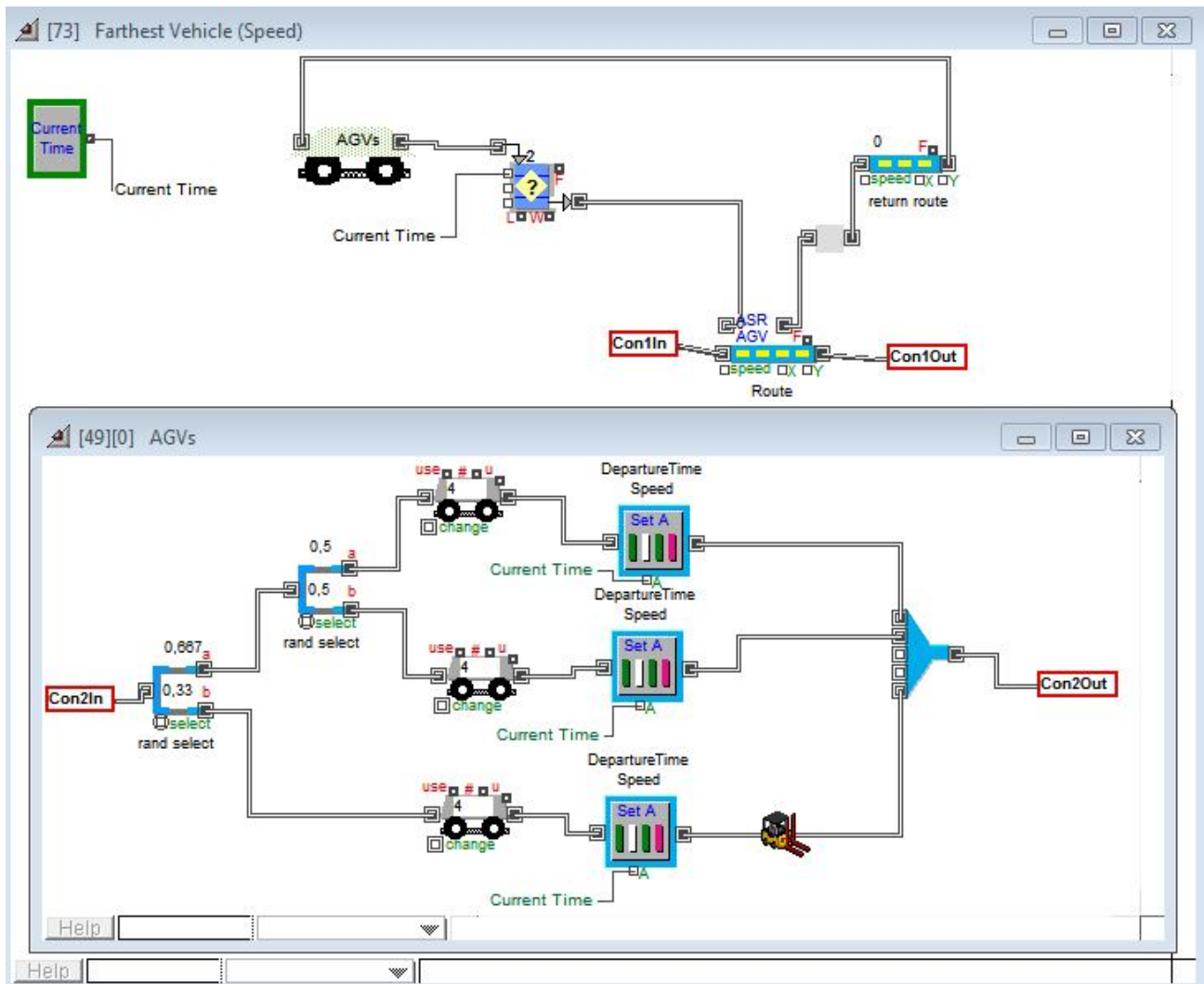
Picture 12g - **The FV (speed) model**

As soon as a job occurs and a vehicle is demanded, AGVs start to move towards the demand point (picture 13g). Since AGV are generated, they pass through the next block in the raw, where they are assigned with two new attributes, totally necessary for the model; speed and departure time. Beginning from the vehicle on top, vehicles' speeds are set 3, 4 and 5 m/s respectively. The departure times are formed depending on the time that the AGV started its route. The described procedure takes place in the "AGVs" hierarchical block. Thereafter, the vehicles exit passing through the "Con2Out" connector and are gathered in the queue, where they wait to be dispatched.

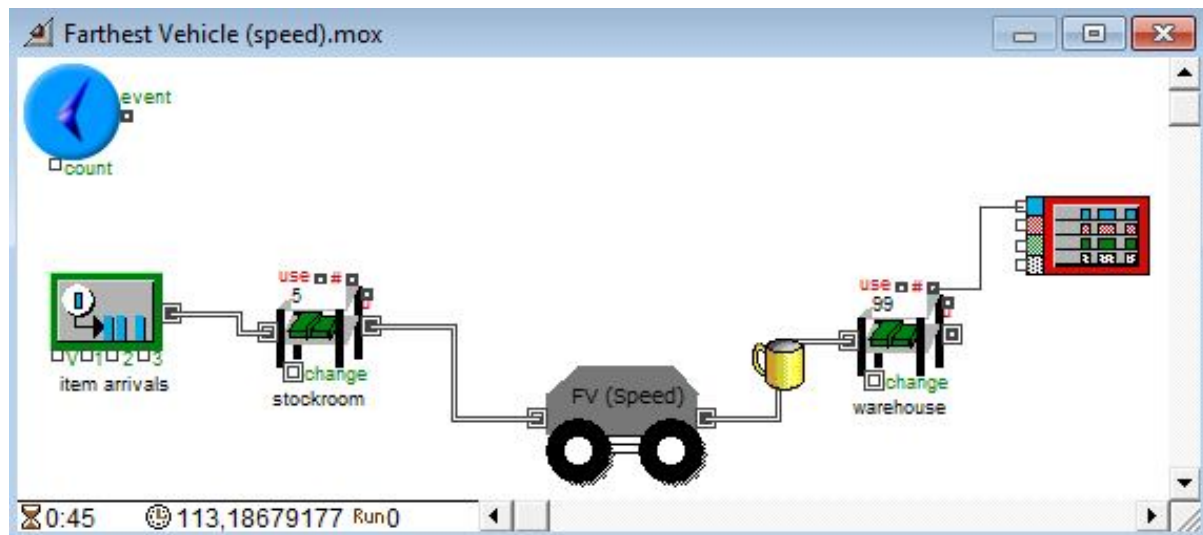
Nonetheless, in order to work properly, it is not enough for the algorithm the vehicles' speed and departure time. This is the reason why "Queue Decision" block uses the values that are formed by the block's connector, which is "Current Time". Thus, the algorithm calculates the travel time of the vehicles (by subtracting departure time from the arrival time in the queue – current time) and multiplies it

with their speed. The AGV with the maximum result is the one that will be released and receive the cup.

Consequently, it receives the cup, transfers it to the labeling machine and set it free to go to the warehouse (picture 14g). This picture also shows that when the time of the simulation is 113,18679 there are 5 unlabeled cups in the stockroom and 99 labeled cups in the warehouse already.

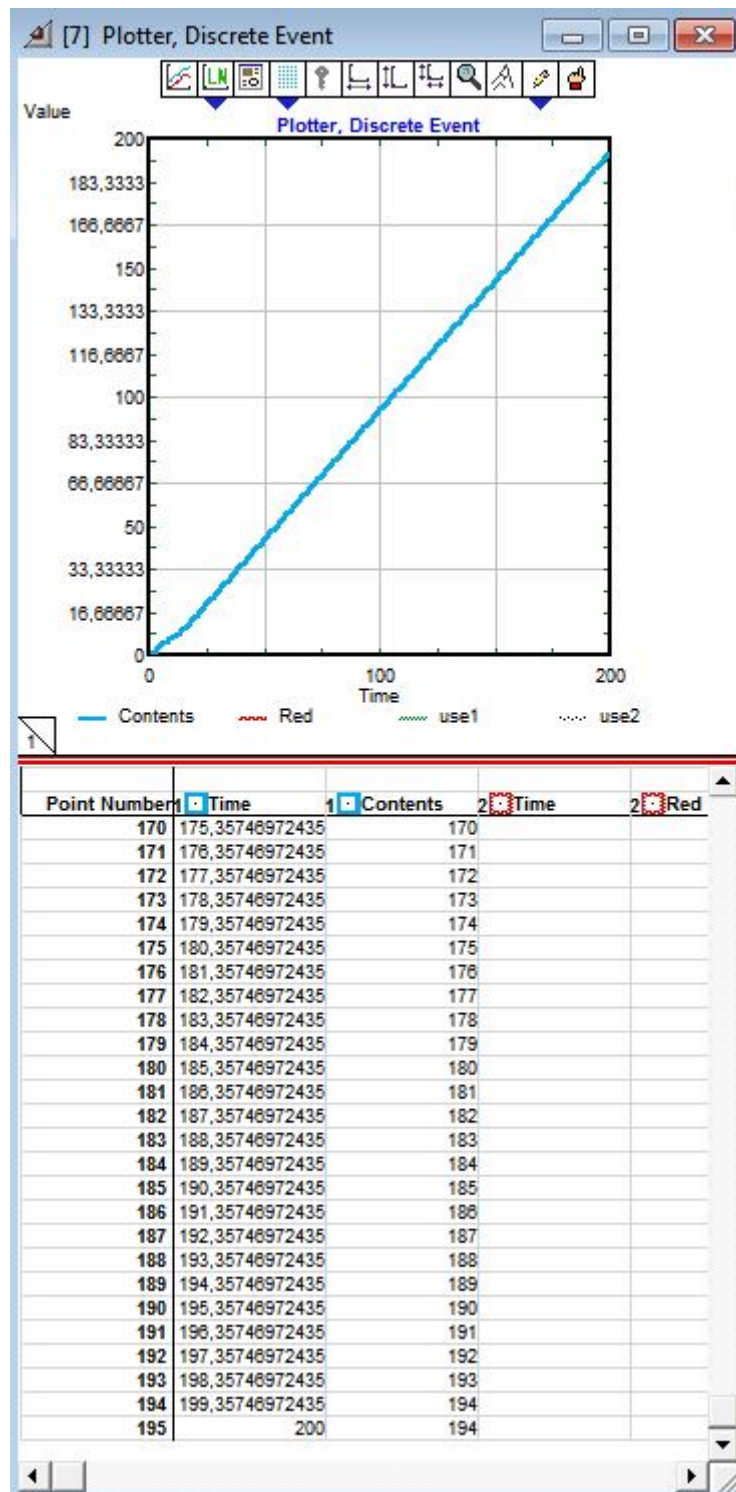


Picture 13g - The "FV (speed)" hierarchical block



Picture 14g - The FV (speed) model

Furthermore, the simulation plotter (picture 15g) presents the times that each labeled cup entered the warehouse in detail. For example, here, it is depicted that when the simulation time is over there are 194 labeled cups in the warehouse.



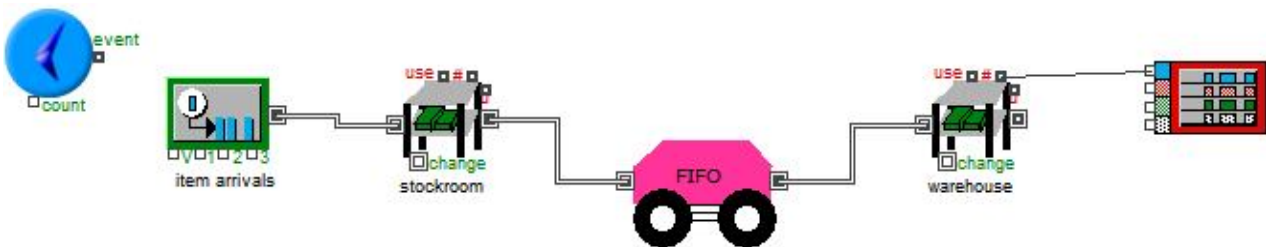
Picture 15g- The simulation diagram

h. First In First Out Vehicle (FIFO)

i. About the FIFO model

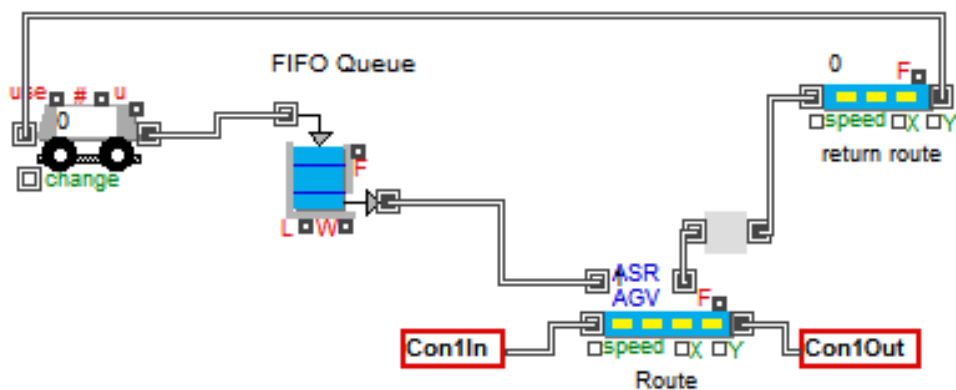
The FIFO model requires that the entering and exiting of the queue will be performed in a FIFO order, i.e. if the vehicle i is assigned for a job before vehicle j, then vehicle i must be dispatched before vehicle j.

Picture 1h presents the FIFO model, which consists of the usual blocks that are mentioned in every model. There is the executive block, the block that generates items, a plotter block and two stock blocks which gather temporarily items. Finally, there is the FIFO hierarchical block, which is the basic block of the model.



Picture 1h – **The FIFO model**

The items enter from the left connector and wait for a vehicle to load them (picture 2h). Several vehicles start to move from their generator block ("AGV" block) directing to the items' route. However, on their way, they pass through the "FIFO Queue" block. This block ensures that the first vehicle that will pass in will be the one that will pass out first too.



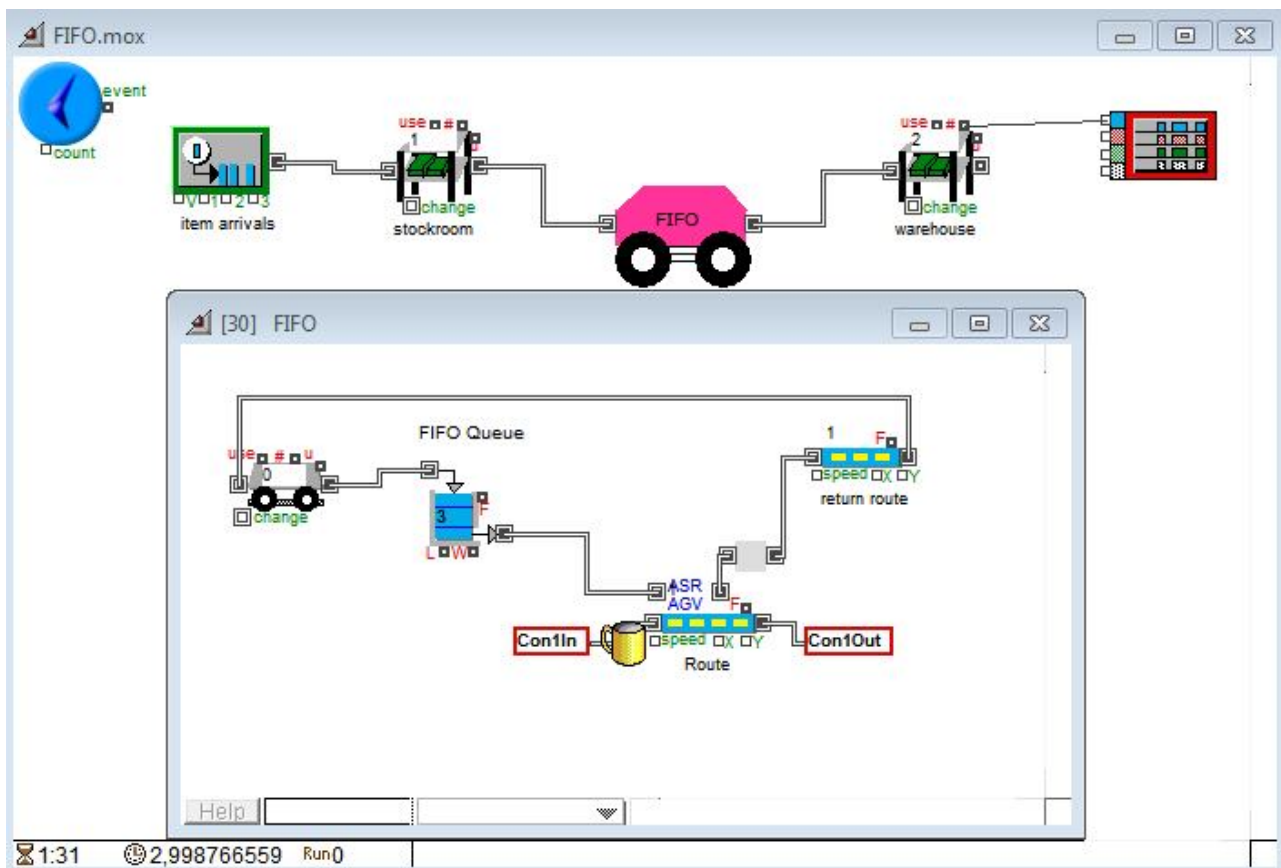
Picture 2h – **The "FIFO" hierarchical block**

Since the vehicle is released, it receives and transfers the item until the end of the route, where it sets the item free from the right connector and returns to its initial position through a return route.

ii. Running the FIFO model

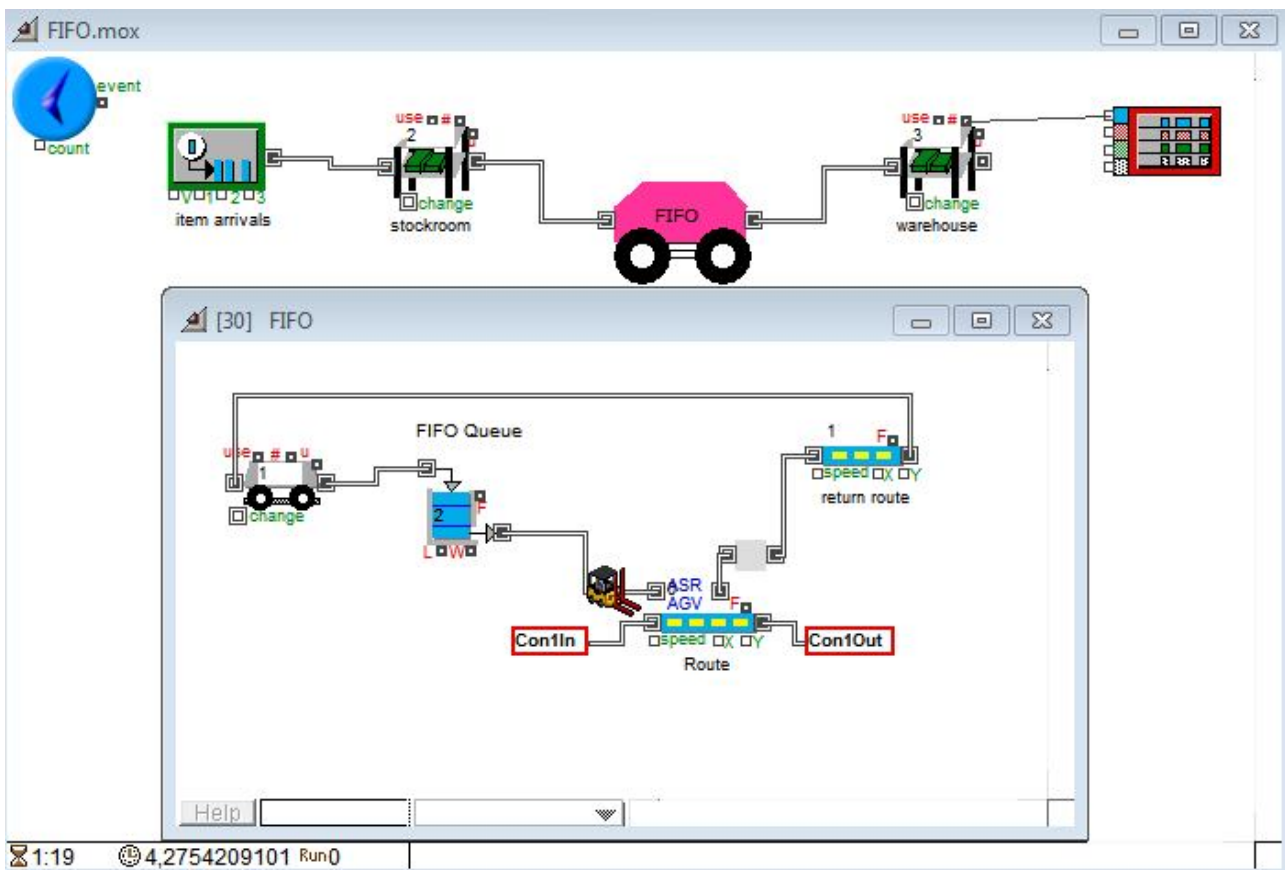
It is assumed that there are 200 cups with no label on them, 200 time units and 5 available AGVs which are assigned to carry out the procedure of transferring the cups to the labeling machine.

When the time is 2,9987 time units, it seems that an unlabeled cup is approaching the "Route" block demanding for an AGV (Picture 3h).



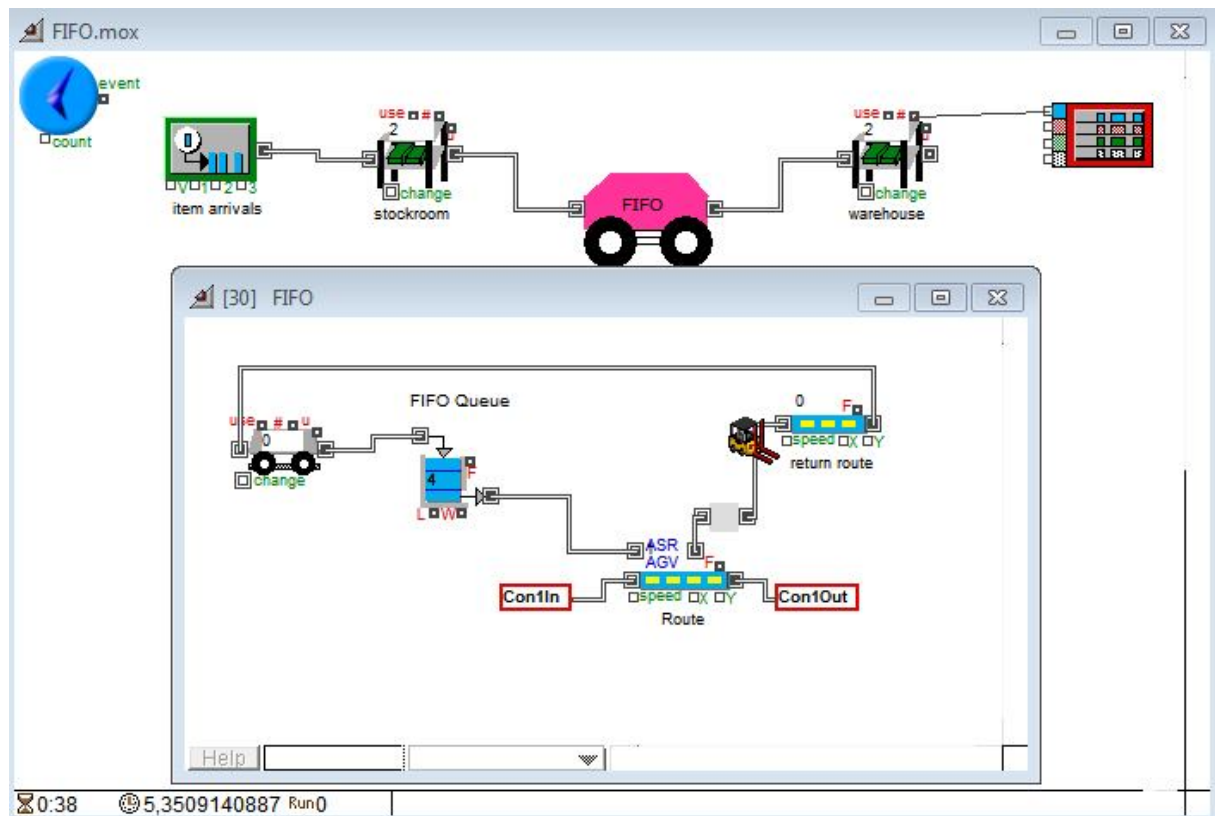
Picture 3h – **The FIFO model**

The first available vehicle reaches the demand point when the time is 4,2754 units (Picture 4h). As soon as the unlabeled cup is loaded to the vehicle, they travel together to the labeling machine, in order to stack the firm label on the bottom of the cup.

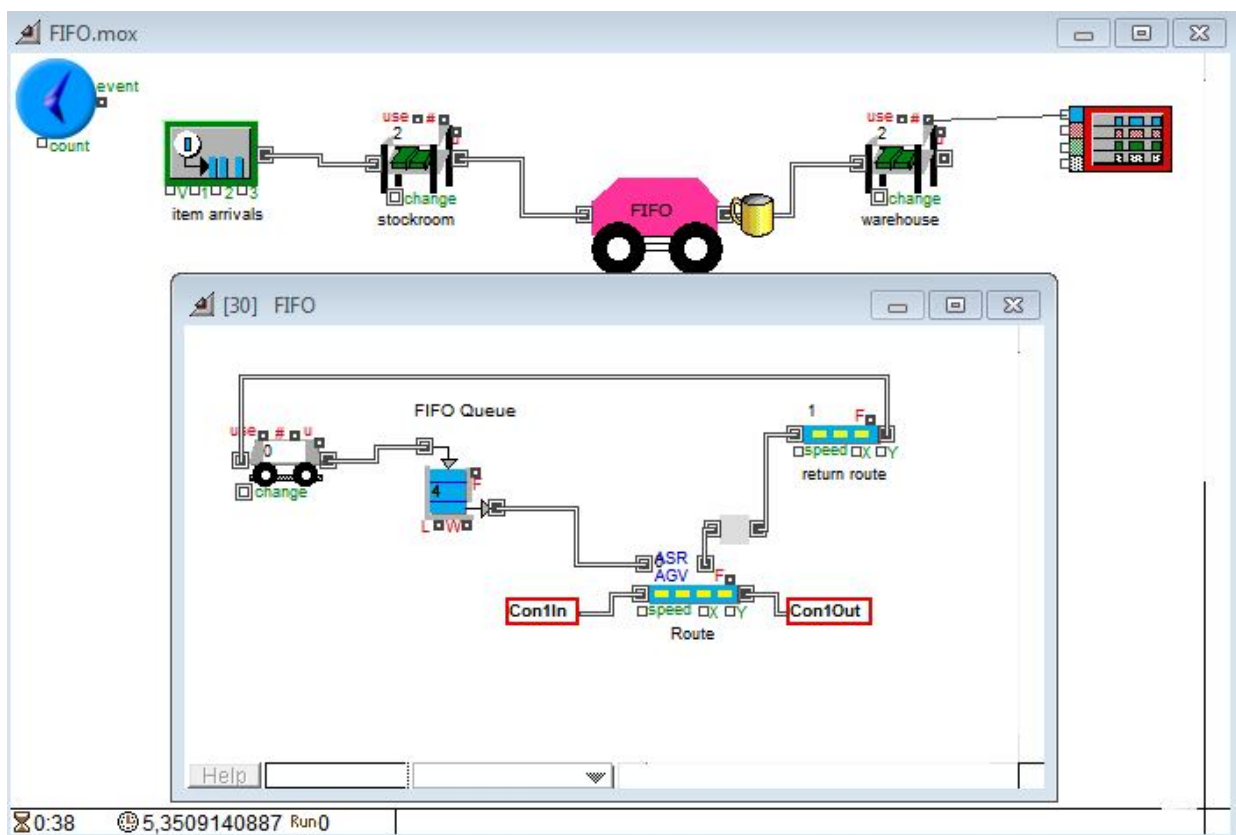


Picture 4h – **The FIFO model**

Since the job is done the labeled cup is released and the AGV returns to its block. The picture below (picture 5h) shows that when 5,3509 time units have elapsed, an empty AGV is on its return route as soon as it has been released, whereas at the same time a labeled cup has just left the hierarchical block and is traveling to the warehouse (Picture 6h).

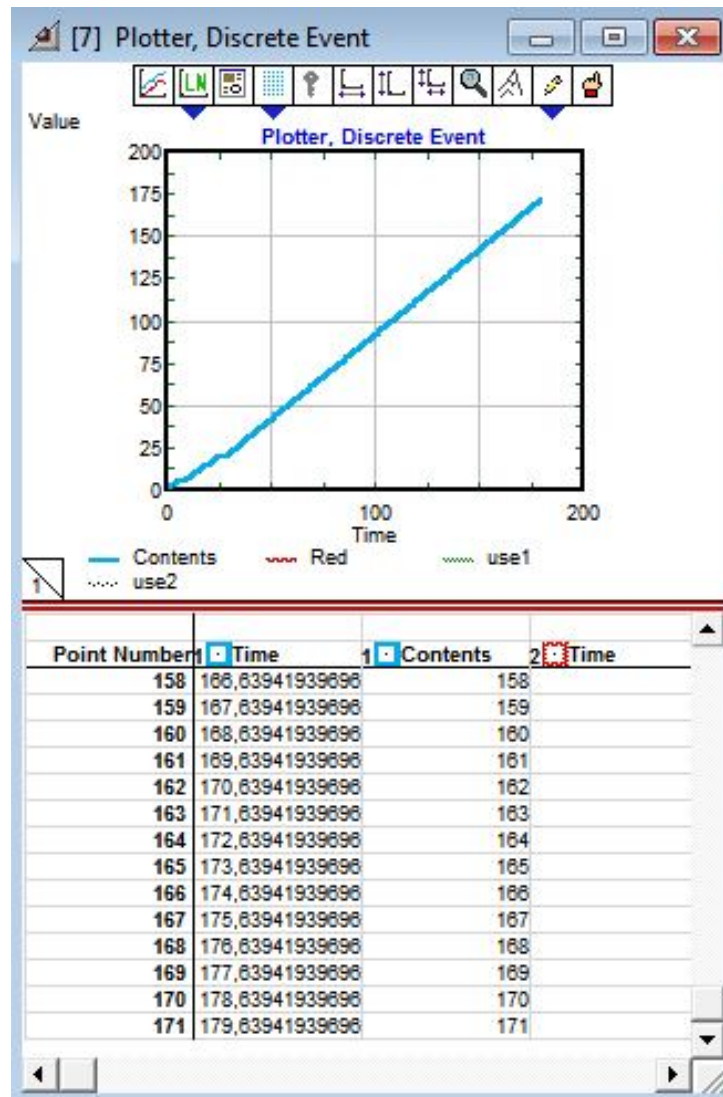


Picture 5h – **The FIFO model**



Picture 6h – **The FIFO model**

At the end, the arrivals of the labeled cups in the warehouse can be presented in a diagram like below. This diagram denotes that for example at time 0 there are no cups in the warehouse, at time 174,63941 there are 166 cups, at time 175,63941 there are 167 and so on.



Picture 7h - The simulation plotter

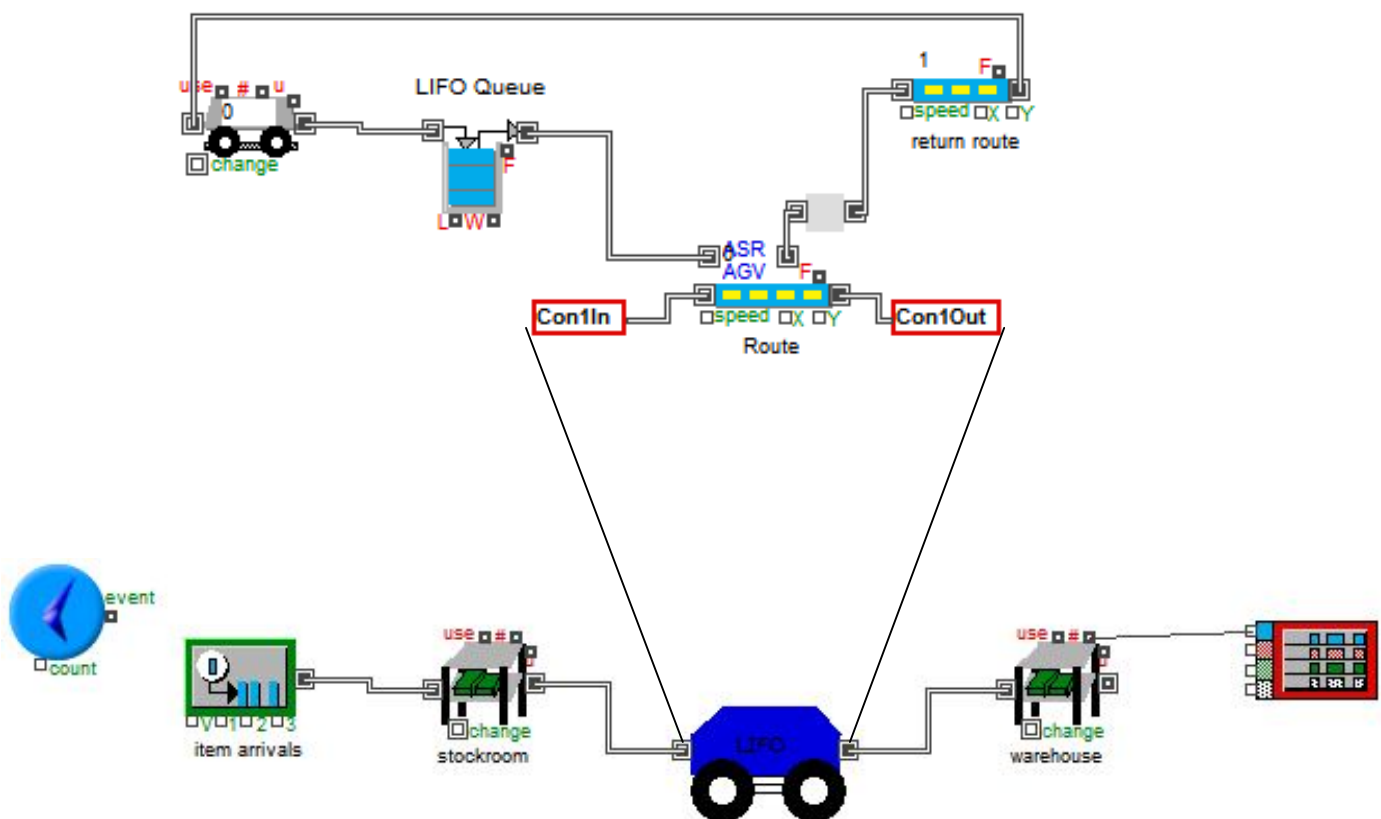
i. Last In First Out Vehicle (LIFO)

i. About the LIFO model

The LIFO model requires that the entering and exiting of the queue will be performed in a LIFO order, i.e. if the vehicle i is assigned for a job before vehicle j, then vehicle i must be dispatched after vehicle j. The order of the vehicles that pass

in and out the queue reminds of a stack. That is, the first vehicle that passes in, is the last that will be dispatched.

Picture 1i presents the entire LIFO model. It has the main blocks in common with the former models apart from the hierarchical block. Furthermore, the “LIFO” hierarchical block has the same block with the “FIFO” except for the block that controls the queue. Instead of the “FIFO Queue”, there is the “LIFO Queue” block that ensures that the last vehicle which enters the queue is the one that will leave first.



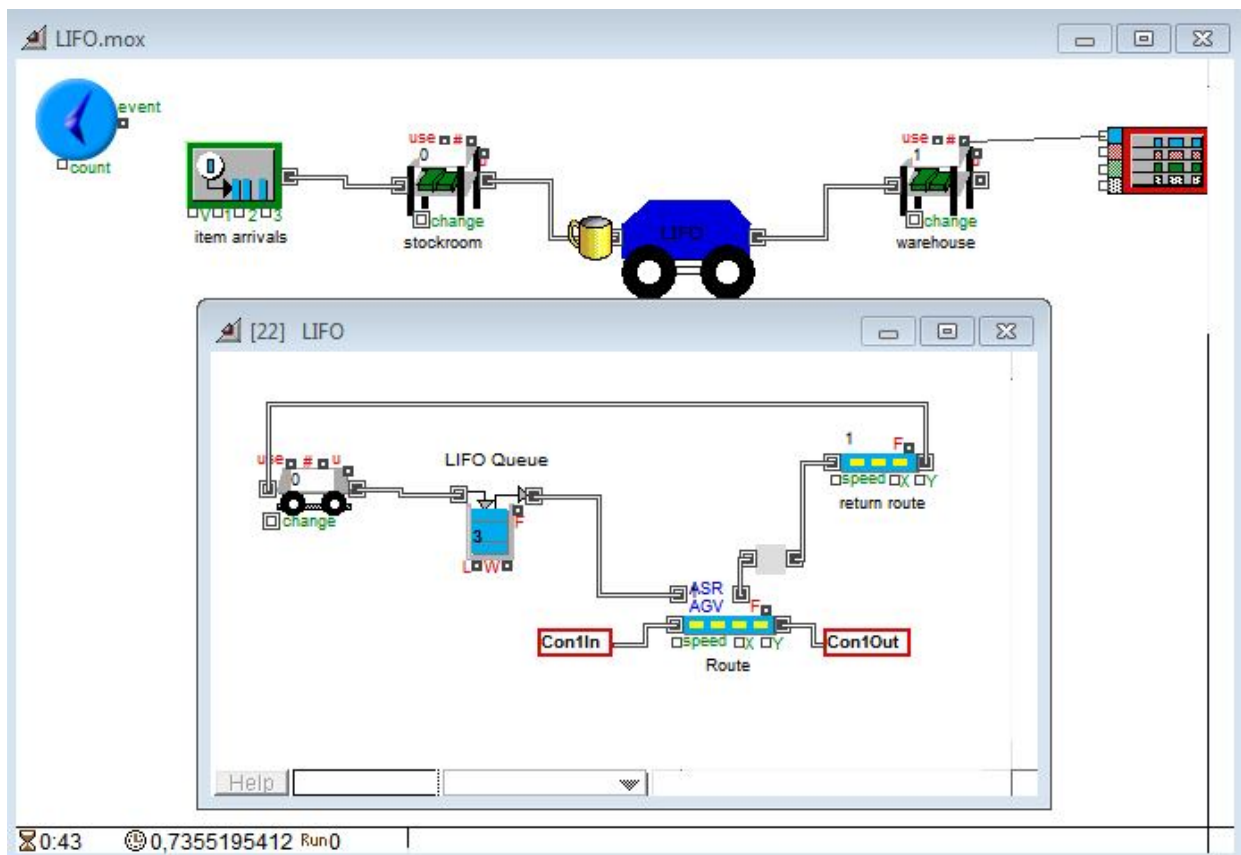
Picture 1i– **The LIFO model**

ii. Running the LIFO model

It is assumed that there are 200 unlabeled cups and 5 AGVs which have to pick up the first ones and transfer them to the labeling machine. For the entire procedure the vehicles have 200 time units available.

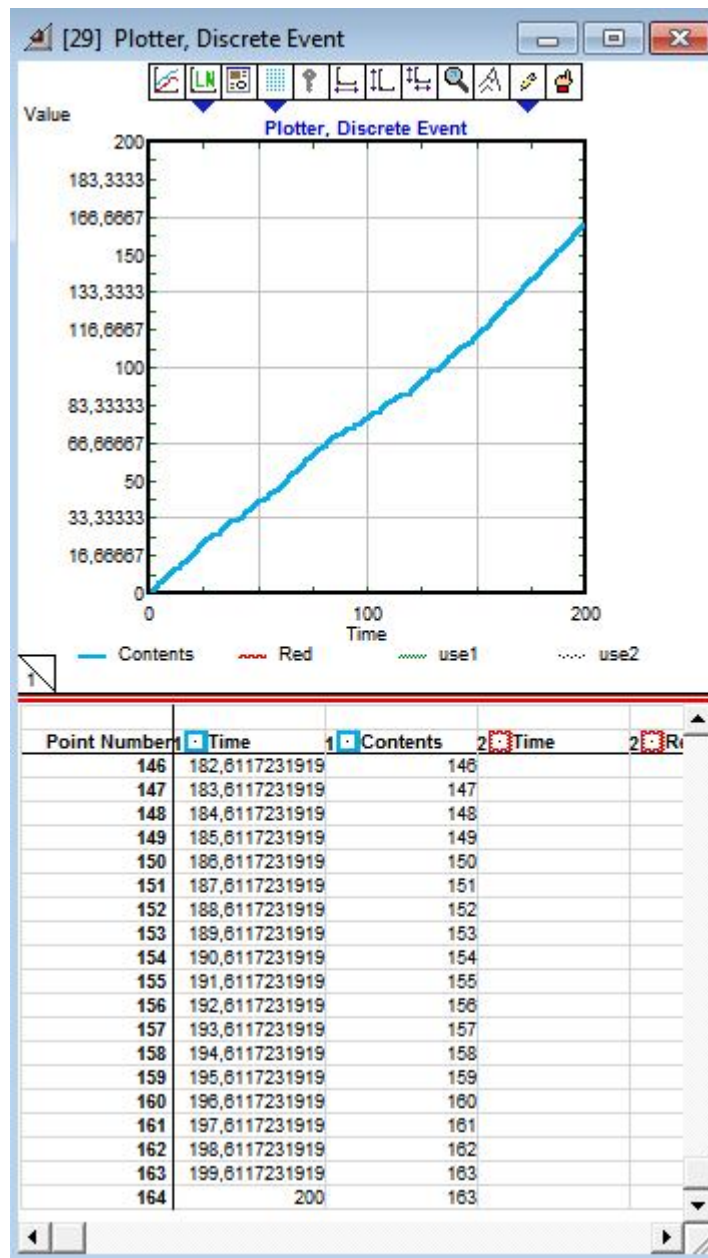
The following picture shows an unlabeled cup entering the “LIFO” hierarchical block at 0,73551 time units. The same time, there are already 3 AGVs waiting in the

line of the queue, whereas there is an empty vehicle that is returning to its block. What is more, it is noticeable that a labeled cup is already in the warehouse.



Picture 2i – **The LIFO model**

At last, the next picture presents the diagram of the labeled cups' arrivals in the warehouse during the simulation. Naturally, the given time to the AGVs was not enough to carry out the procedure for 200 cups, yet only 163 cups were labeled.



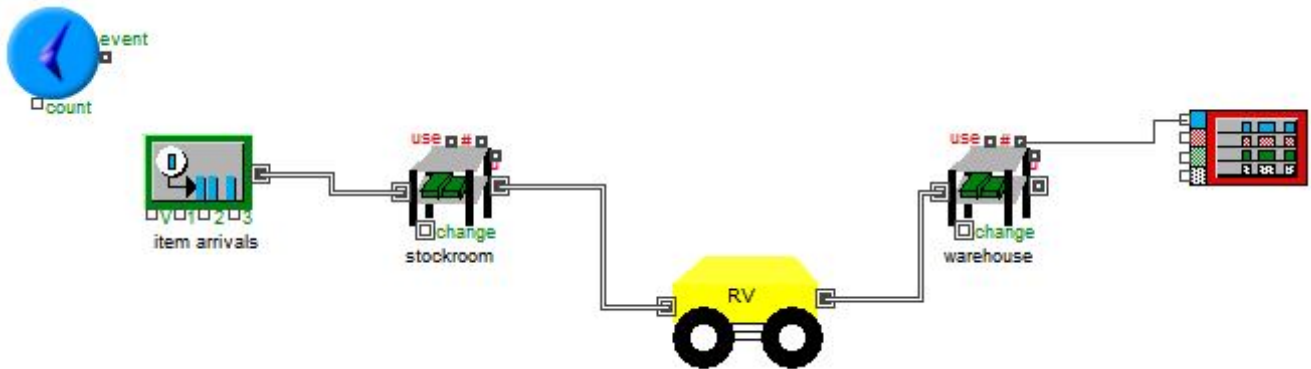
Picture 3i - The simulation plotter

j. Random Vehicle (RV)

i. About the RV model

According to this model each vehicle is randomly chosen to complete the task. The selection is made without taking into account any specific characteristic of the AGVs.

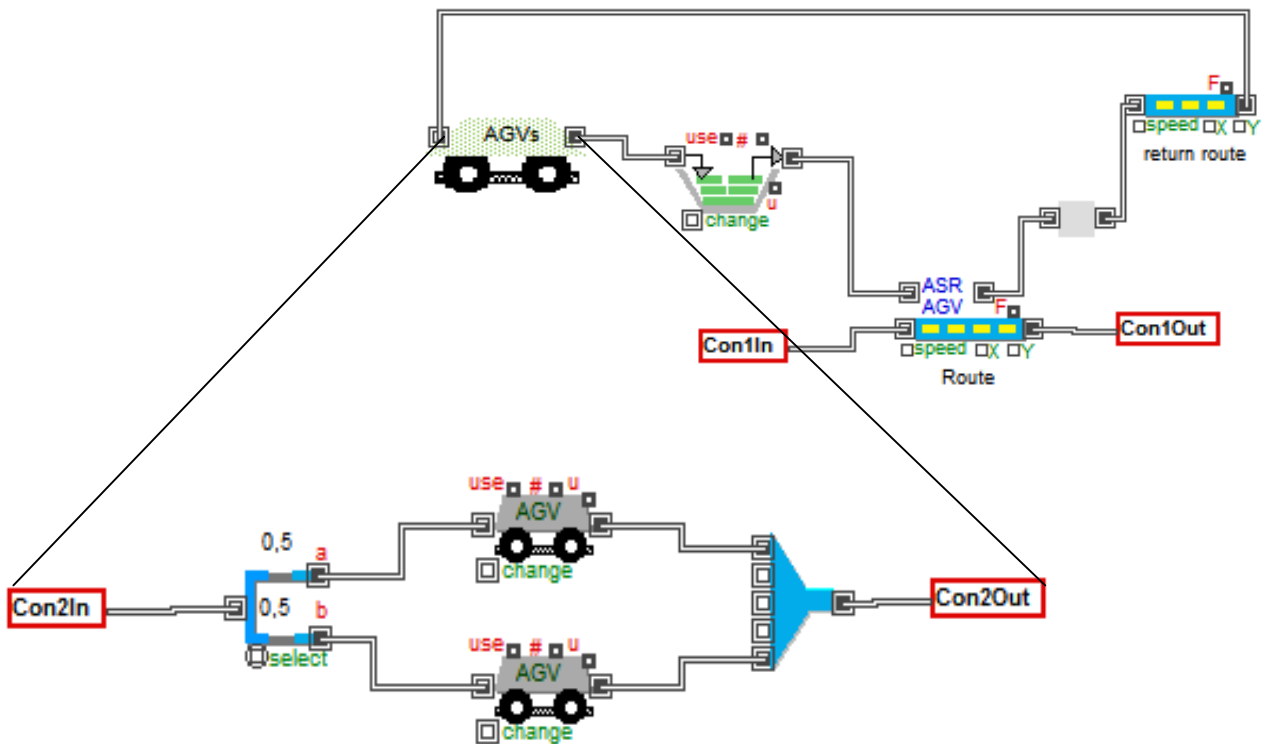
As far as the view of the model concerned, is the same with the rest models. Thus, there is the executive block, the generator block, two stock blocks, the plotter and finally the hierarchical one, RV (Picture 1j).



Picture 1j – **The RV model**

The picture below (Picture 2j) shows the inside of the “RV” block. To begin with, there is a hierarchical block (“AGVs”) that consists of “AGV” blocks. It can consist of any number of this type of block, each of which has one or more different characteristics or attributes. When the vehicle passes out this hierarchical model through the “Con2Out” connector it runs into a bin. The bin provides a storage location for collecting and retrieving items, without guaranteeing a specific order for the collection or retrieval.

Consequently, when there is an item waiting for a vehicle at the route, it is picked any vehicle from the bin without regarding its attributes.

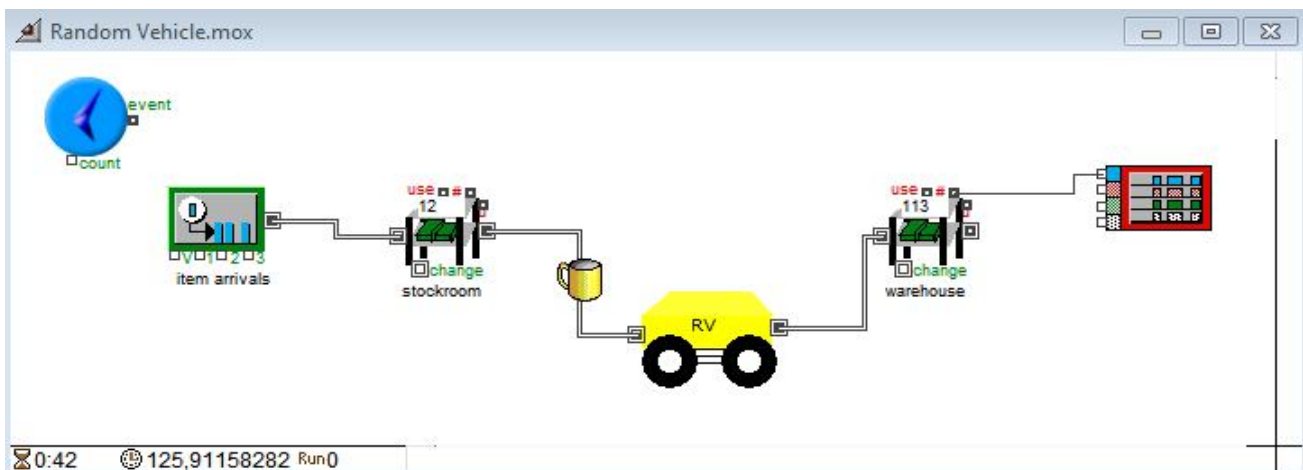


Picture 2j – The "RV" hierarchical block

ii. Running the RV model

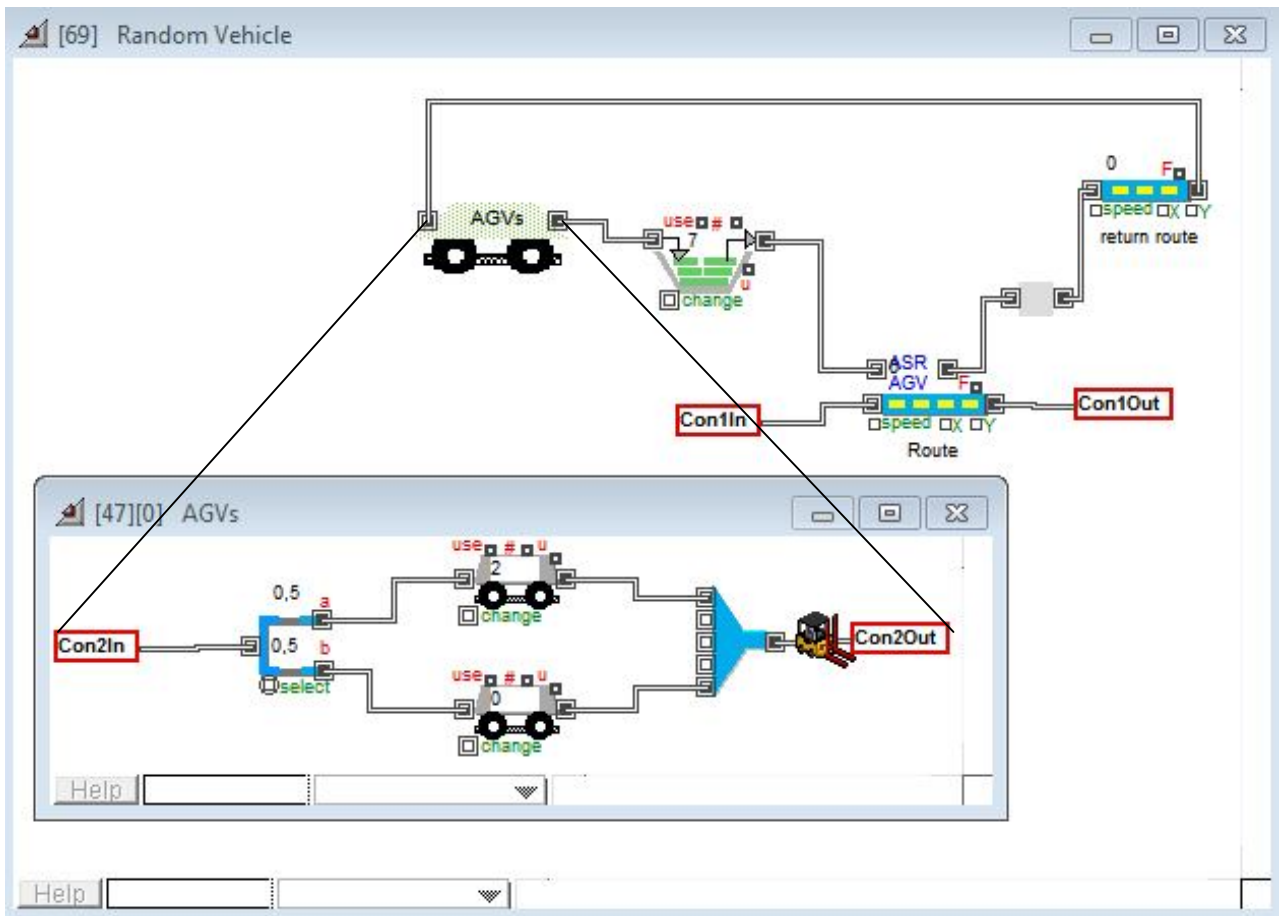
It is assumed that there are 200 unlabeled cups and 10 AGVs which have to pick up the first ones and transfer them to the labeling machine. For the entire procedure the vehicles have 200 time units available.

The following picture shows, that at 125,91158 time units, there are 12 unlabeled cups waiting at the stockroom, an unlabeled cup entering the "RV" hierarchical block and 113 labeled cups in the warehouse.



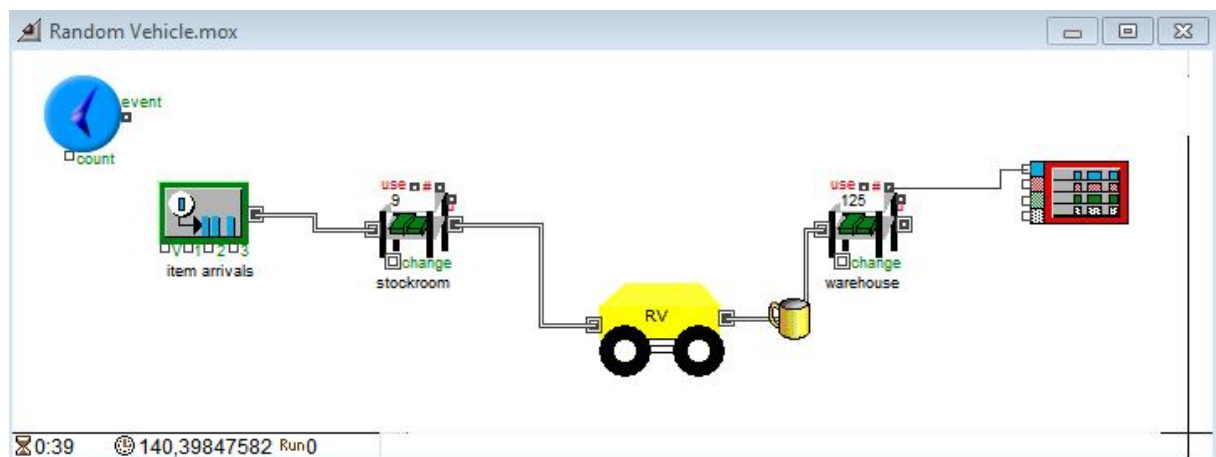
Picture 3j – The RV model

What is more, at the same time, a vehicle is passing out the “AGVs” hierarchical block and is ready to enter the bin, where there are already 7 vehicles. As soon as there is a vehicle demand, an AGV is picked randomly from the bin. None of the vehicles have priority against another.



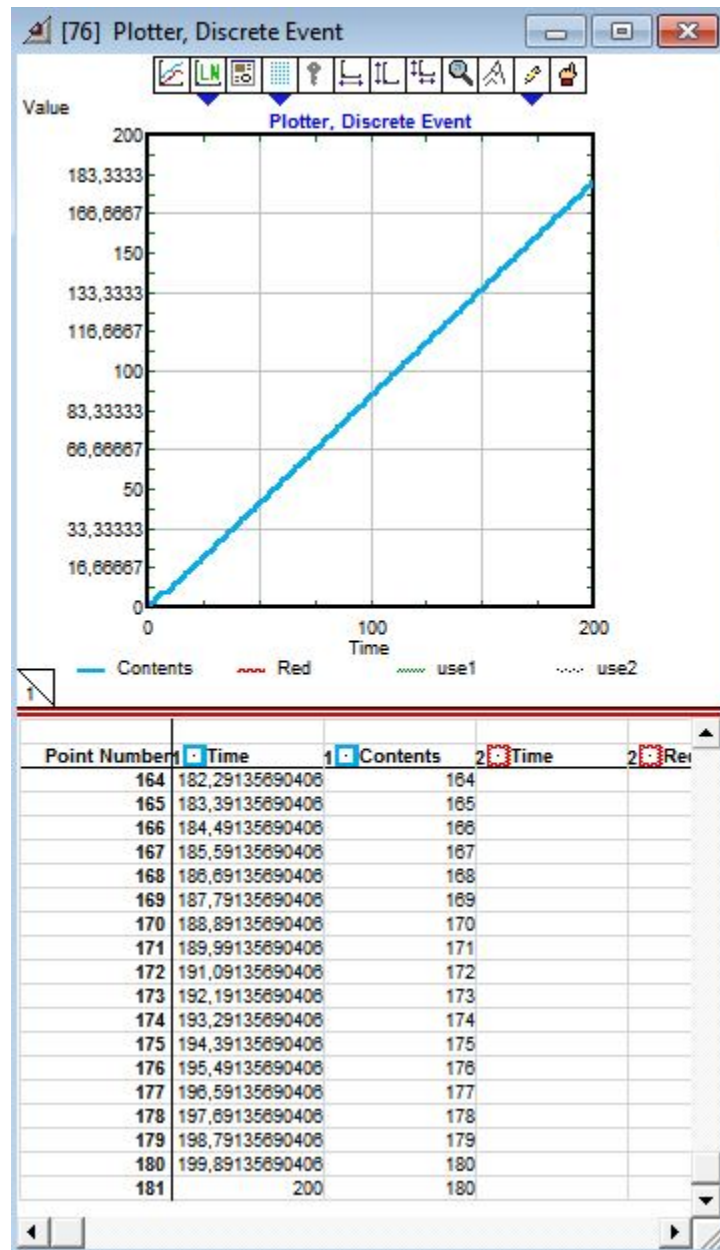
Picture 4j – The "RV" hierarchical block

After the selection, the vehicle arrives to receive the cup, transfers it to the labeling machine and set it free to go to the warehouse (Picture 5j). This picture also shows that when the time of the simulation is 140,39847 there are 9 unlabeled cups in the stockroom, yet 125 labeled cups in the warehouse.



Picture 5j – The RV model

At last, the next picture presents the diagram of the labeled cups' arrivals in the warehouse during the simulation. As it is logical, the given time to the AGVs was not enough to carry out the procedure for 200 cups, yet only 180 cups were labeled.



Picture 6j - **The simulation plotter**

7. Conclusion

A flexible manufacturing system consists of machines, where production operations are performed, a material handling system, which links the vehicles with the work stations and a central computer control. In contrast to a transfer line where all parts follow the same sequence of operations, the material handling system permits the parts to follow a variety of different routings (Egbelu, 1987). Flexible manufacturing systems demand flexible material handling systems and automatic guided vehicles are a state of art of this type of handling systems.

The development of the technology brought to the firms two significant results. The first one regards the variety of alternatives solutions for the transportation in a plant that accrued, whereas the second one is related to the fact that material handling systems can be more easily implemented in these plants.

An Automated Guided Vehicle system is a type of material handling system that consists of a number of driverless vehicles (AGVs), which travel on a guide path and are controlled by a central computer (Hammond, 1986).

Automated Guided Vehicle systems are material handling devices that already are in widespread use and their importance for material handling is expected to grow rapidly. The advantages that such systems can offer include increased flexibility, better space utilization, improved factory floor safety, reduction in overall operating cost, and easier interface with other automated systems. Apart from the numerous advantages that these vehicles can offer though, there are some disadvantages that are stressed. Thereafter, it is pointed out how an AGV system can be set up, taking into consideration both design (physical elements such as the vehicles and the facility layout) and control (dispatching, scheduling and routing a vehicle).

At a primary stage, the aim of this study was to provide information regarding flexible manufacturing environments and automatic guided vehicles which are essential part of them. Additionally, there is a literature review related to design, vehicle scheduling, dispatching and routing. However, the main purpose of the study was to analyze the several aspects of modeling and simulation of AGV systems.

Modeling is a useful tool which can represent, analyze or even evaluate real systems which are often really complex. This is the reason why, combined with simulation, it gains more and more attention in both academic and practical level (Kouskouras, Georgiou, 2007). The detailed depiction of reality enables somebody to observe thoroughly a model as if it was the real system. It can be observed how the real system reacts under several conditions, and various hypotheses can be tested at a fraction of the cost of performing the actual activity. What is more, when observing

a model is easier to think of solutions in order to improve the real-world system, rather than observing the real one.

The model use has many advantages such as reduced costs and shortened design cycles. One of the major benefits of a model though, is that it can be at first a simple approximation of a process and gradually be enriched with more elements imitating better the real system.

Tools used in AGV system design can be classified in two main categories: analytical tools and simulation based tools. Analytical tools are mathematical techniques such as queuing theory, integer programming, heuristic algorithm, and Markov Chains, whereas simulation-based tools can be grouped in three categories (Tanchoco, 1994):

1. General-purpose simulation languages (e.g. SLAM II, SIMAN IV, etc.)
2. Simulation packages designed for the general simulation of manufacturing systems (e.g. SIMPLE++, AutoMod II, ProModel, SIMFACTORY II.5, etc.)
3. Simulation software specially created for analyzing AGV systems by using general programming languages such as C, FORTRAN, BASIC, LISP, etc.

ExtendSim software uses ModL as its internal programming language. When the blocks are built in Extend using ModL, the block's program is compiled to native machine code. ModL is essentially C with a few extensions and changes. Practically, this software is used in order to create a model to simulate a real world system under certain conditions. Thus, it is made use of the simulation's advantages by reducing both cost and risk.

Subsequently, the extended use of models was a motivation of creating a set of models that will simulate the basic scheduling rules for the Automated Guided Vehicles systems. Thus, new hierarchical blocks were built, in order each of which to represent one different scheduling rule. All in all, there were built twelve hierarchical blocks for the twelve scheduling rules. These are: Least Idle Vehicle, Least Cumulative Idle Vehicle, Longest Idle Vehicle, Longest Cumulative Idle Vehicle, Least Utilized Vehicle, Nearest Vehicle-distance, Nearest Vehicle-speed, Farthest Vehicle-distance, Farthest Vehicle-speed, First-In-First-Out Vehicle, Last-In-first-Out Vehicle and Random Vehicle.

Every single block is a subsystem of the main system that includes the relevant algorithm, which eventually decides the vehicle that will be dispatched. It is independent from the system, which means that it can be used individually in any system of any simulation that it is desirable to be applied one of the former vehicle scheduling rules. Due to this practicality of the hierarchical blocks, it was regarded

useful to integrate them in a library, named "Hierarchical". Thus, models can be constructed without even typing in an equation.

It is important to note that the simulation model focuses more on the behaviors of its components rather than the entire system's behavior. What is more, the number of the real world's details that will be introduced in the model depend on the respective capabilities of each simulation software.

Before the implementation of the AGV systems in the industries, their planning and design should be thoroughly studied, as the possibilities of an implementation failure may be high. Because of their high cost, space and maintenance requirements, there is no question that AGV systems must be evaluated before their implementation if their benefits are to be fully realized.


Due to their complex, dynamic behavior and high degree of interaction with all the other components of the manufacturing system, it is difficult to characterize their performance analytically (Kuhn 1983). Hence, alternatives to simulation modeling of material handling systems do exist. Operations research methodologies, particularly queuing network models have been used to study a number of performance measures such as machine/station utilization, throughput rates, machine queue data, work in progress, etc. Their primary advantage is speed of use, deterministic output, which can be easily interpreted, and low cost due to their limited data input and computer processing requirements.

In order to use AGVs though, it is often necessary to make unrealistic assumptions. These assumptions may involve deterministic service times or unlimited space in the buffers. Another significant disadvantage of AGVs is that there are some issues that cannot be studied. These issues are regarded as vehicle blocking interactions, failures' effects etc. With the availability of simulation software and lower cost computers, queuing network models have lost much of their attraction, in spite of being fast, easy and inexpensive to use.

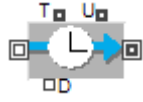


Hence, along with their increasing use, design and planning problem of AGV systems has been a major concern of study. Simulation is widely used to control and evaluate the system performance for AGV systems. Many attempts have been made for developing AGV simulators either for specific problems or as generic-applicable to any AGV system simulators. However, there is still a strong need for a generic and extensible (by using object oriented programming approach) AGV simulator. Thus, future research efforts can be directed to create such simulation systems.

Appendix

Executive – needed in every model

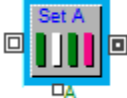
Library Blocks	Function
 Executive	<p>This block is the heart of each discrete event model and must be placed to the left of all other blocks in the model. It allows the duration of the simulation to be controlled by the end time or by another number specified in the dialog. Generally you will have no reason to change the default values in the dialog.</p>

Activity – Processing items

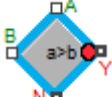
Library Blocks	Function
 Activity Delay	<p>Holds an item for a specified amount of delay time, then releases it. The delay time is the value in the dialog or, if connected, the value at the D connector when the item is received (the connector overrides the dialog).</p> <p>This block can be used for any kind of service delay. For example, you can use this block to represent red lights in traffic, the time it takes a clerk to wait on a customer, or multitasking CPU time.</p>
 Route	<p>Simulates a route for either AGV and ASR vehicles and the items they transport. The vehicles are pulled into the route through the AGV/ASR input connector, while items to be transported enter through the unlabeled item input connector. The unlabeled item output connector holds items to be picked up by other blocks (such as another Route or a batch block).</p> <p>If the vehicle is to acquire items at points along the route, several routes can be cascaded with a batch block between each route. Routes may be cascaded by connecting the unlabeled item output of one route to the unlabeled item input of the next.</p> <p>Once a vehicle is on a route, succeeding Route blocks will recognize it until it is released back to a vehicle block. The block at the end of a vehicle route should have the "Free vehicle at end" option selected in the dialog to push the vehicle out the AGV/ASR output connector. The AGV/ASR output connector is then connected to the input connector of the originating vehicle block. A vehicle will be freed up to return to its resource block regardless of whether or not the item remains stuck at the output connector because the downstream path is blocked.</p> <p>Multiple vehicles may be allowed on the route at the same time. You can set the maximum number of AGVs on the route at once in the dialog.</p>
 Route Delay	<p>Simulates a route for AGV and ASR vehicles when they are not transporting items (such as before pickup or after drop off). Unlike the Route block, this block is not used for transporting items, only for delaying a vehicle, such as when the vehicle is returning to the source. The input and outputs are</p>

	<p>vehicles, not items.</p> <p>Route Delay blocks can be cascaded by connecting the output of one to the input of the next. You can connect a Route Delay block at the beginning or end of a regular Route block.</p>
--	---


Attributes

Library Blocks	Function
 <p>Set Attribute</p>	<p>Sets the attributes of items passing through the block. Up to seven attribute names and values may be assigned to an item with each Set Attribute block. The attributes may add to or replace existing item attributes. You can specify the value of one of the attributes with the A connector. The value at the A connector overrides the corresponding value in the dialog.</p>


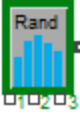
Decisions – Routing on deciding which value to use

Library Blocks	Function
 <p>Decision</p>	<p>This block makes a decision based on the inputs and internal logic you define. The dialog lets you perform the following tests comparing A to B: greater than, greater than or equal to, equal to, less than, less than or equal to, and not equal. You can also test for A being an invalid number (noValue).</p>

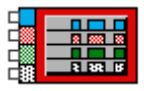
Generators – Creating items, scheduling

Library Blocks	Function
 <p>Generator</p>	<p>Provides items for a discrete event simulation at specified interarrival times. Choose either a distribution on the left, or choose the empirical distribution and enter probabilities in the table. Items can be created with a random distribution or at a constant rate of arrival. You can also specify the number of items output at each event in the dialog or at the V connector.</p> <p>This block provides items at specified interarrival rates. Since it always pushes items, this block should usually be followed by a Queue or Resource block when used to provide items for the model. Otherwise, you may lose some items that are generated. If an arrival rate of 0 or less occurs, items are generated immediately (at the time the 0 or less value occurs).</p> <p>The parameters for the distribution arrival times are set in the dialog. The random distributions include: beta, binomial, constant, empirical, Erlang, exponential, gamma, hyperexponential, log normal, normal, Pearson type V, Pearson type VI, Poisson, Triangular, uniform integer, uniform real, and Weibull. The empirical distribution may have up to 20 points and may be interpreted as a discrete, stepped, or interpolated distribution.</p>


Input/Output – Reading and writing files, or generating values

Library Blocks	Function
 System Variable	Allows you to regulate some aspect of the model based on the status of the simulation. It is usually used in conjunction with a decision-type block, for example, to halt a process after current time reaches a certain value. The variables you can use are: current run number, current step, current time, end time, number of runs, number of steps, start time, and time step.
 Input Random Number	Generates random integers or real numbers based on the selected distribution. You can use the dialog or the three inputs, 1, 2, and three to specify arguments for the distributions. You can select the type of distribution: Uniform (integer or real), Beta, Binomial, Erlang, Exponential, Gamma, Geometric, HyperExponential, LogLogistic, LogNormal, Neg. Binomial, Normal, Pearson type V, Pearson type VI, Poisson, Triangular, Weibull, and Empirical. The Empirical distribution uses a table of up to 50 values to generate a discrete, stepped, or interpolated empirical distribution.


Plotters

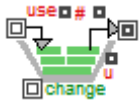

Library Blocks	Function
 Plotter	This block provides a plotter depending on the inputs.

Queues – Holding, sorting and ranking items

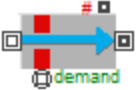

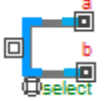
Library Blocks	Function
 Queue Decision	This queue releases items according to the results of the primary equation defined on the Queue tab. When the queue can release an item downstream, the equation is evaluated for each item in the queue. Depending of the setting of the "Release items with..." popup menu, the item with the highest, lowest, or first TRUE value is released. If multiple items have the same result value, the secondary equation defined on the Tie Break tab is evaluated to determine which item is to be released. If there secondary equation is left blank, the item which has been in the queue the longest will be released.

Resources – Representing items as resources

Library Blocks	Function
 AGV	Provides auto-guided vehicles for use with the Route block. The AGVs are output into the simulation one at a time. Feed the AGVs back to the block when they reach the end of their route. This block is used with the Route block that pulls the required AGVs from the output connector. The Route block determines the path and speed of the vehicles, and returns those not in service to the AGV block. If you want to model the time it takes for the AGV to return to the block, use a Route (Delay) block. Note that, unlike other resource blocks, this block and the ASR block do not allow you to add default

	attributes and priorities.
	Provides a storage location for collecting and retrieving items such as parts and tool bits. Use the change connector to modify the number of items in the bin in cases such as when a parts lot has gone bad.
 Stock	Provides and stores stockroom items such as raw materials, work in process, and so on. This block may be used in an open system such as when items are shipped, or in a closed system such as when you exchange parts in spares inventory. Items can be pulled from the output as long as there are items available. You can add attributes and priorities to items passing through the block.

Routing – Moving items to the correct place

Library Blocks	Function
 Activity Service	Passes an item only when the demand connector is connected and certain conditions exist at the demand input (either demand's value is true [greater than 0.5] or it pulls in an item). Depending on the type of output connector (item or value) attached to demand, this block passes single items or passes a specified number of items. This block allows service on demand. You can think of this block as a path with a gate that opens on demand, where demand can accumulate.
 Combine	Combines the items from two different sources into a single stream of items. This is different from the batch blocks which join items from several sources into one item. The items in the Combine block retain their separate identities and are not batched together.
 Select Output	Selects the input item to be output at one of two output connectors based on a decision. The item at the input is passed through the selected output. The dialog has options for changing the outputs after a given number of items have passed and selecting based on the select connector.

References

Akturk, M. S., Yilmaz, H., (1996), 'Scheduling of Automated Guided Vehicles in a Decision Making Hierarchy', *International Journal of Production Research*, Vol. 34, No 2, 577-591

Ben Saoud, S., Jaoua, A., Bellamine-Ben Saoud, N., (2005), 'Efficient Simulator Based on Meta-Heuristic for FMS and AGV Systems Design and Control', *Journal of Computer Science* (Special Issue), Vol. 8-14

Brazier, M.K., Shannon, R.E., (1987), 'Automatic Programming of AGVs Simulation Models', *Industrial Engineering Department*, Texas A&M University

Buzacott, J. A. , Shanthikumar, J. G., (1980), 'Models for Understanding Flexible Manufacturing Systems', *IIE Transactions*, Vol.12, No 4, 339-350

Buzacott, J. A., David, D. Yao, (July, 1986), 'Flexible Manufacturing Systems: A Review of Analytical Models', *Management Science*, Vol. 32, No 7, 890-905

Christofides, N., Eilon, S., (1969), 'An Algorithm for the Vehicle-Dispatching Problem', *Operational Research Society*, Vol. 20, No 3, 309-318

Davis, D., (1986), 'Modeling AGV Systems', *Proceedings of the 1986 Winter Simulation Conference*

De Koster, R. (M.)B.M., Le-Anh, T., J. van der Meer, R., (2004), 'Testing and classifying vehicle dispatching rules in three real-world settings', *Journal of Operations Management*, Vol. 22, 369–386

Desrochers M., Desrosiers J., Solomon M., (1992), 'A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows', *Operations Research*, Vol. 40, No 2, 342-354

Diamond, B., Lampertis, S., Krah, D., Nastasi, A., (2002), *Extend v6*, Imagine That, Inc.

Duinkerken, M.B., Ottjes, J.A., Lodewijks, G., (Dec, 2006), 'Comparison of Routing Strategies for AGV Systems using Simulation', *Simulation Conference, WSC 06, Proceedings of the Winter*, 1523-1530

Egbelu, P. J., (1987), 'Pull Versus Push Strategy for Automated Guided Vehicle Load Movement in a Batch Manufacturing System', *Journal of Manufacturing Systems*, Vol. 6, No 3, 209-221

Egbelu, P. J. ,Tanchoco, J. M. A., (1984), 'Characterization of automatic guided vehicle dispatching rules', *International Journal of Production Research*, Vol. 22, No 3, 359-374

Ganesharajaha, T., Hall, N. G. , Sriskandarajah, C., (1998), 'Design and operational issues in AGV-served manufacturing systems', *Annals of Operations Research*, Vol 76, 109-154

Ganesharajaha, T., (1997), 'AGV-Served Manufacturing Systems: Scheduling and Design in Loop Layouts', Phd , Mechanical and Industrial Engineering Department, University of Toronto

Gendreau, M., Hertz, A., Laporte, G., (1994), 'A Tabu Search Heuristic for the Vehicle Routing Problem', *Management Science*, Vol. 40, No 10, 1276-1290

Giffler, B. , Thompson, G. L., (1960), 'Algorithms for Solving Production-Scheduling Problems', *Operations Research*, Vol. 8, No 4, 487-503

Graves, S. C., (1981), 'A Review of Production Scheduling', *Operations Research*, Vol. 29, No 4, 646-675

Kimemia, J., Kimemia, G., Stanley, B., (1983), 'An Algorithm for the Computer Control of a Flexible Manufacturing System', *IIE Transactions*, Vol.15, No 4, 353-362

Kouskouras K.G., Georgiou A.C., (2007), 'A discrete event simulation model in the case of managing a software project', *European Journal of Operational Research*, Vol 181, 374-389

Kuhn, A., (1983), 'Efficient Planning of AGVS by Analytical Methods', Proceedings of the 2nd International Conference on Automated Guided Vehicle Systems, *IFS Publications Ltd.*, 1-10

M. van Heijden, Ebben, M., N. van Gademann, A. van Harten, (2002), 'Scheduling vehicles in automated transportation systems; Algorithms and case study', *OR Spectrum*, Vol. 24, 31–58

Mahadevan, B. and Narendran, T. T., (1994) 'A hybrid modeling approach to the design of an AGV based material handling system for an FMS', *International Journal of Production Research*, Vol. 32, No 9, 2015 — 2030

Mohsen, A., (1992), 'Flexible Manufacturing Systems', *Information Systems Management*, Vol.9, No 2, 44 — 47

Naish, D. M., Croft, A. E., Benhabib, B., (2003), 'Coordinated dispatching of proximity sensors for the surveillance of maneuvering targets', *Robotics and Computer Integrated Manufacturing*, Vol. 19, 283–299

Qui, L.L., Wen-Jing H., Shell-Ying H., Wang H., (2002) 'Scheduling and routing algorithms for AGVs: a survey', *International Journal of Production Research*, Vol. 40, No 3, 745-760

Reveliotis, S. A., (2000), 'Conflict resolution in AGV systems', *IIE Transactions*, Vol. 32, No 7, 647-659

Ross, E.A., Mahmoodi, F., Mosier, C. T., (1996), 'Tandem Configuration Automated Guided Vehicle Systems: A Comparative Study', *Decision Sciences*, Vol. 27, No 1

Sarin, S.C. and Chen, C.S., (1987), 'The machine loading and tool allocation problem in a flexible manufacturing system', *International Journal of Production Research*, Vol. 25, No 7, 1081-1094

Sezen, B., (2003), 'Modeling Automated Guided Vehicle Systems in Material handling', *Üniversitesi Dergisi*, Vol.4, No 2, 207-216

Shneor, R., Berman, S., Edan Y., (2006), 'Fuzzy Dispatching of Automated Guided Vehicles', *Applied Soft Computing Technologies: The Challenge of Complexity Advances in Soft Computing*, Vol. 34, 749-760

Solomon, M., (1987), 'Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints', *Operations Research*, Vol. 35, No 2, 254-265

Taghaboni-Dutta, F., Tanchoco, J. M. A., (1995), 'Comparison of dynamic routing techniques for automated guided vehicle system', *International Journal of Production Research*, Vol. 33, No 10, 2653-2669

Tanchoco, J.M.A. (1994) *Material Flow Systems in Manufacturing* , Chapman & Hall.

Ujvari, S., Hilmola, O., (2006), 'Advanced manufacturing simulation: Minor system details can be major issues in the real world', *Industrial Management & Data Systems*, Vol. 106, No 8, 1166-1186

Robotics Online, viewed on 10 July 2010, <<http://www.robotics.org>>