Computer Science and Systems Analysis
Technical Reports

Miami University *Year* 1994

Intelligent Simulation Modeling of a Flexible Manufacturing System with Automated Guided Vehicles

Onur Dulgeroglu
Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & SYSTEMS ANALYSIS

TECHNICAL REPORT: MU-SEAS-CSA-1994-001

Intelligent Simulation Modeling of a Flexible Manufacturing System with Automated Guided Vehicles

Onur Dulgeroglu

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

Intelligent Simulation Modeling of A Flexible Manufacturing System With Automated Guided Vehicles

by

Onur Dulgeroglu
Systems Analysis Department
Miami University
Oxford, Ohio 45056

Working Paper #94-001 May, 1994

SYSTEMS ANALYSIS DEPARTMENT

MASTER'S PROJECT FINAL REPORT

Presented in Partial Fulfillment of the Requirements
for the Degree of
Master of Systems Analysis
in the
Graduate School of Miami University

TITLE: Intelligent Simulation Modeling of a Flexible Manufacturing System with Automated Guided Vehicles

PRESENTED BY: Onur Dulgeroglu

DATE: April 21, 1994

COMMITTEE MEMBERS:

Donald Byrkett

James Kiper

Mufit Ozden, Advisor

Alton Sanders

Intelligent Simulation Modeling of a Flexible Manufacturing System with Automated Guided Vehicles

Onur Dulgeroglu

Department of Systems Analysis
Miami University, 1994

ABSTRACT

Although simulation is a very flexible and cost effective problem solving technique, it has been traditionally limited to building models which are merely descriptive of the system under study. Relatively new approaches combine improvement heuristics and artificial intelligence with simulation to provide prescriptive power in simulation modeling.

This study demonstrates the synergy obtained by bringing together the "learning automata theory" and simulation analysis. Intelligent objects are embedded in the simulation model of a Flexible Manufacturing System (FMS), in which Automated Guided Vehicles (AGVs) serve as the material handling system between four unique workcenters. The objective of the study is to find satisfactory AGV routing patterns along available paths to minimize the mean time spent by different kinds of parts in the system. System parameters such as different part routing and processing time requirements, arrivals distribution, number of palettes, available paths between workcenters, number and speed of AGVs can be defined by the user. The network of learning automata acts as the decision maker driving the simulation, and the FMS model acts as the training environment for the automata network; providing realistic, yet cost-effective and risk-free feedback.

Object oriented design and implementation of the simulation model with a process oriented world view, graphical animation and visually interactive simulation (using GUI objects such as windows, menus, dialog boxes; mouse sensitive dynamic automaton trace charts and dynamic graphical statistical monitoring) are other issues dealt with in the study.

Table of Contents:

I. INTRODUCTION

I.A. Traditional Simulation Modeling

I.A.1. Advantages of Simulation Modeling over Analytical Approaches

Simulation is a powerful problem solving technique. It can be used to experiment with systems which are not yet in existence, or with existing systems without actually altering the real system; and therefore offers valuable reductions in terms of time, cost, and risk involved in modeling systems, designing experiments and playing scenario analysis games.

Although simulation analysis is limited in some aspects, its popularity as a decision making aid is increasing in direct relation to the capability and accessibility of today's high speed digital computers. Computer simulations are assuming the role of traditional experiments in many areas of business and scientific investigations as coding and running simulation models of large, complex real life systems (both in the manufacturing and service sectors) is becoming more and more profitable with the improving technology.

Generally, the real life systems we analyze are composed of closely interconnected sub-systems. There are various -seemingly independent- sources of information and multiple points of decision making. What is more, *randomness* is a very important, non-negligible factor in life: real systems are usually hierarchical, distributed, and contain a large number of relatively independent, but still implicitly coordinated decision makers operating under great uncertainty.

The complexity of real world problems are such that in a lot of cases, the simplifying assumptions made by the corresponding analytic model(s) might not be realistic, or the appropriately formulated model cannot be *solved* analytically.

When the uncertainty encountered in a system is sufficiently small, existing analytical methods can be suitably modified to cope with them: In fact, many of the algorithms dealing with stochastic systems are closely related to their counterparts in deterministic systems. However, when uncertainty is large, modifying existing algorithms is not enough: new paradigms have to be considered to take care of the random environment, and simulation modeling is a very promising alternative to capture the real stochastic behavior of the system under study.

I.A.2. Shortcomings of Simulation as a decision making tool

The most important drawback of traditional simulation analysis is that although it can capture the dynamics and randomness of a system to the level of detail desired and *describe* it successfully, it does not *suggest* the optimal decisions to be made. It does not provide a *solution* to potential or existing problems, and therefore does not *directly* help improve the system under study.

Simulation is considered to be a method for developing Descriptive Models: They merely describe the system behavior and leave the optimization process totally to the analyst. This can be an important drawback, since most systems involve a huge number of scenarios to choose from, and many possible actions that can be taken at any point while searching for an acceptable near-optimal solution. The limitations of the mere descriptive approach will become more apparent after the FMS under consideration and the objectives of the study are discussed.

Two simulation experts say "...but even within the simulation discipline, which treats **only descriptive** models, a large number of possibilities exist[1]." Simulation *is* a valuable tool when used in the traditional way, but it is growing to be even more powerful, since it **is *not* limited to building descriptive models** any more.

---

[1] Hoover and Perry, 1990.

I.B. "Intelligent" Simulation Modeling

This study demonstrates the **synergy** obtained by using "intelligent" objects in simulation modeling. Combining learning automata theory and simulation analysis, a network of intelligent objects are built into the model in order to go beyond the traditional *descriptive* limitations of simulation analysis. As the model runs, the decision policies randomly produced by these intelligent objects evolve in view of the feedback they get from the stochastic environment, and converge to certain values that will result in better states of the system: intelligent, or *prescriptive* simulation modeling.

I.B.1. What others have done

Benjaafar (1992) points out the fact that simulation analysis has proven to be superior to traditional mathematical modeling in capturing the scope, complexity and dynamics of manufacturing systems, and as a result, has become increasingly popular in the design and control of manufacturing systems in the last few years. However, "...it remains merely a descriptive modeling tool. This leaves the user with the difficult tasks of generating, testing, and selecting design and operation alternatives." As an improvement, a new architecture based on a "structured object representation paradigm" is suggested, and a methodology that describes the supplementation of simulation systems with more intelligent and prescriptive functions is proposed. The author also mentions the design of an experimental prototype system that allows automatic model generation and simulation, as well as model evaluation.

In (Potter, 1990), the authors introduce the design and prototype implementation of an "Active Knowledge/Data Language (KDL)" which provides an integration of model bases, knowledge bases, and data bases. They assert that the Active KDL can be used to provide a very powerful intelligent modeling environment.

In (Snyder, 1988), the authors explore the problem of efficiently combining a non-deterministic decision capability with simulation models, in the context of a model implemented in a discrete-event simulation language (SIMAN). Their purpose was to find areas of synergy between simulation and artificial intelligence, and introduce the idea of a "decide node" which would help the analyst in "bridging the gap between data and wisdom".

In (Rao, Jiang, Tsai, 1990) , (Rao, Jiang, Tsai, Chang, 1987) , (Rao, Ying, Jiang, Psai, 1990) and (Rutan, 1989), the authors have worked toward creating intelligent simulation environments with the help of expert systems, artificial intelligence and simulation synergism.

Although it is clear that other researchers have realized the power of using "intelligent simulation models", there are very few examples that are close to the approach described in this study: using a series of interconnected dynamic, *learning* decision making units to bring an evolving, self-convergent nature to the key decisions made during a *single* simulation run.

I.C. The Learning Automaton

The Learning Automata Theory is a relatively new method to analyze systems, bring an explanation to failures, and extend successful results. "The underlying learning approach implements adaptive decision making in highly uncertain stochastic environments. In addition, the learning automata can be implemented in a distributed manner, a critical feature for dealing with large, complex systems.[2]"

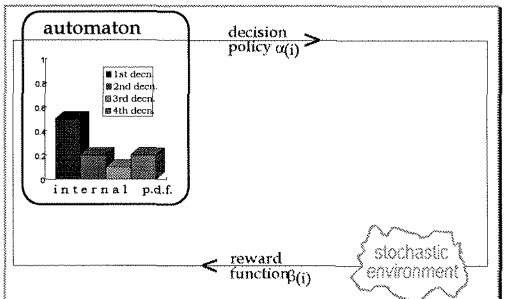


Figure 1: A learning automaton improves its decisions by interacting with the stochastic environment.

The decision makers (Figure 1) operate in a random environment from which they receive only probabilistic feedback (a reward function) regarding the effect of the action taken.

A learning automaton makes decisions based on its discrete probability distribution. The decision is then executed in the stochastic environment, and a reward function is generated by making use of some quantifiable performance measures. There are several ways of updating the probability distribution of a learning automaton by using the feedback from the system. In this study, the **"reward-inaction"** learning algorithm will be used. A brief description of this algorithm is presented in Exhibit 1.

REWARD - INACTION LEARNING ALGORITHM:

β = reward function, in the range [0,1].
(Decisions that make the system behave worse are not punished. They are just rewarded with $\beta=0$.)
i = decision that has resulted in reward β_i
P [ ] = discrete probability distribution array
a = learning constant
(theoretically in the range (0,1], but usually ≤ 0.1)

$$P[i] \leftarrow P[i] + a \cdot \beta_i \cdot (1 - P[i])$$
$$P[k] \leftarrow P[k] - a \cdot \beta_i \cdot P[k] \ , \ \forall k \neq i$$

Since 1-P[i] equals Σ Pk, $\forall k \neq i$ for a probability density function that is valid to start with, the increase in P[i] is compensated by the cumulative decrease in others, and validity of the p.d.f. is not lost.

Exhibit 1

A general purpose simulation environment would be ideal to develop applications of the learning automata theory because of the synergy obtained: "The major weakness of the general simulation approach as an analysis and problem solving tool is that it lacks the power in prescribing the satisfactory policies to decisions involved in a complex system. On the other hand, the major weakness of the learning automata is that they need a long training period and dependable, realistic feedback to be able to converge to satisfactory policies.[3]"

In this study, the network of learning automata represents the distributed decision making units, that is, the "brain" driving the simulation model. On the other hand, the simulation model acts as the learning environment for the automata, with built-in conveniences such as automatic data collection, statistical analysis, visually interactive modeling and graphical user interface.

The result is an "intelligent simulation model" with prescriptive power. This "learning" approach to simulation is brand new, and is quite exciting for both the people who have been using simulation in the traditional descriptive way, and those who will *become* simulation users with this prescriptive dimension introduced.

Although this study concentrates on using the learning automata theory to help converge to near-optimal solutions in AGV routing patterns, potential utilization of the methods outlined here is *not* limited to this particular type of problems. Similar learning algorithms can be implemented to aid any kind of decision making process with a quantifiable objective measure that can be transformed into an appropriate reward function for the network of learning automata.

---

2 Narendra and Thathachar, 1989.
3 Ozden.

II. THE PROBLEM AREA

II.A. Flexible Manufacturing Systems (FMS) and Automated Material Handling

Flexible Manufacturing Systems (FMSs) are production settings which combine the efficiency required for medium to high volume production, yet still preserve the flexibility of a job-shop environment for *concurrent* production of a variety of products. Automated material handling systems are usually a must for achieving required levels of throughput. Automated Guided Vehicles (AGVs) are the latest trend in advanced FMSs in the recent years: the high mobility of these vehicles increases flexibility without sacrificing speed.

Establishing an AGV system is a complex and expensive decision. Although careful design and detailed operational planning are essential, there are not very many analytical tools to help the decision makers in laying out the network configurations and control policies. Considering the facts that most FMSs have characteristics unique to themselves and that they work in highly stochastic and dynamic environments, it is unreasonable to expect general-purpose analytical tools or mathematical models that would prescribe the optimal solutions for all cases.

This reasoning leads us to the fact that simulation analysis will continue to be a very valuable, cost effective design and justification tool for FMSs with AGVs. After using balancing and path network configuration heuristics, the analyst/designer has to use **design of experiments** and **simulation modeling** to come up with more realistic evaluations of the system under construction.

Simulation is *not* the miracle answer to all the questions either: Even after the analytical tools reduce the solution space to a reasonable subspace, it usually is *still* infeasible to try out all remaining combinations of system parameters in the search for optimal decisions.

First of all, the problem is multi-dimensional. There are too many parameters to experiment with, and too many degrees of freedom in the system configuration solution space: machine scheduling, AGV dispatching to requests from workcenters depending on varying part routing requirements, number of AGVs and number of palettes are some important ones.

Another very important factor affecting the performance of AGV systems is the selection of the path patterns that AGVs will use to travel between workcenters, minimizing delays caused by attempting to use the same track segment in conflicting directions, yet keeping the distance to be traveled to a bare minimum at the same time. Usually, many different track network allocation states are possible, corresponding to different permutations of bi-directional track segments forming the network.

II.B. An Overview of the FMS Under Consideration and Objectives of the Study

II.B.1. Introduction

The FMS that this study concentrates on consists of four workcenters connected with a track network of bi-directional, multi-capacity track segments (Figure 2). The first workcenter is a loading/unloading workcenter (WC#1) and the other three are unique workcenters (WC#s 2 through 4, clockwise) that parts menufactured by the system have to go through to be processed. AGVs are used to carry parts between workcenters.

Our aim is to come up with satisfactory patterns on the track network while routing AGVs, so as to minimize the weighted average time spent by different types of parts in the system, hence increase the effective throughput rate.

The arrival frequency of each part type serves as its weight while calculating the weighted average for the generic flow time Hence, more importance is given to increasing the production rate of parts that are needed more frequently.

II.B.2. Operation of the FMS under study

General behavior:

Parts are assumed to arrive individually to the loading/unloading workcenter: As soon as a palette becomes idle and the state of the loading/unloading workcenter (WC#1) becomes available, a part of random type is created (using the percent arrival distribution as the p.d.f.) and loaded on the palette by WC#1. After visiting all the required workcenters in its routing requirements list, it eventually comes back to WC#1 to be unloaded from the palette, and exits the system.

After a loaded move (carrying their loads to its next immediate destination) the AGVs leave their parts into the input buffers of WCs upon arrival, unless the buffer is full to its capacity. Otherwise, they wait with the part until the input buffer becomes available. After an empty move (to service a requesting WC), AGVs immediately pick their parts up, since the part would already be waiting for the AGV in the output buffer.

The loading/unloading workcenter (WC#1) gives higher priority to unloading parts from palettes (and ship them out of the FMS into the automata network to be evaluated) than introducing new parts to the system. This is usually more beneficial to increase the overall flow, since AGVs are scarce resources and this policy will *occasionally* (when input buffer #1 was full to start with) set the bringing AGV free sooner for it to respond to other requests.

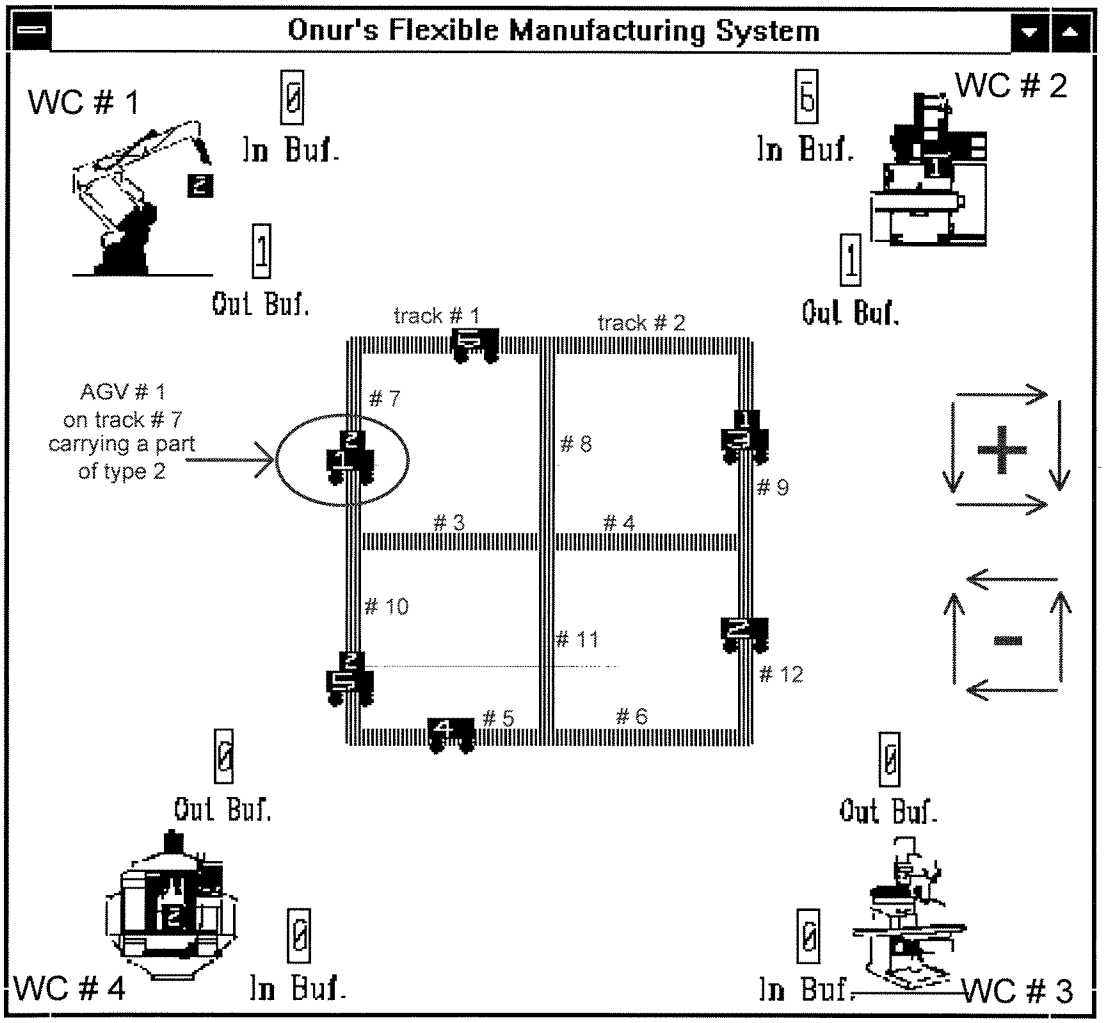


Figure 2: A snapshot of the animation window.

WC#1 does not attempt to introduce a new part to the system when its output buffer is full. Instead, it stays idle until either a part arrives to be unloaded from the palette or a part is taken away from its output buffer, so that starting to work on a new part does not involve the risk of being blocked and *not* being able to do any more loading or especially *un*loading until the output buffer becomes available again. This is a precaution taken against system lock-ups as a result of AGVs not being able to unload their loads, hence never being able to respond to new transportation requests being generated.

All other WCs (#s 2 through 4) attempt to put the part to their output buffer upon completion of the processing time required by the respective part type. If the output buffer is full to capacity, they are blocked until an AGV arrives to pick up a part from the output buffer to carry it to its next destination. The request for an AGV is not made until the part makes it to the output buffer.

Queuing a request as soon as processing is completed might have provided a higher priority to frequently blocked workcenters than those which usually have space in their output buffers for parts waiting to be transferred. However, it would cause some complications in more general cases (for example, the case of different types of parts requiring different kinds of forklifts or AGVs to be transported), since an AGV responding to the *latter* request can arrive at the workcenter *before* the AGV that is moving to respond to the *former* request.

WC#1 has an input buffer of capacity 9, and an output buffer of capacity 5 by default. For WC #s 2 through 4, the input buffers have capacity 9, and the output buffers have capacity 1. Hence these workcenters have a higher risk of being *blocked* (not being able to transfer their part to the output buffer, even though it is finished being processed, hence not being able to process the next part) when compared to the loading/unloading workcenter.

Upon receiving an AGV request, the closest AGV among the idle ones at the time the request is made is selected to respond to that request. If none are idle then the request is queued, and AGVs respond immediately as they become available on a First Come First Served basis.

An AGV can travel in two states:

1. empty: traveling to respond a transport request from a WC with a non-empty output buffer
2. full: carrying the part to its next destination (next WC in its routing requirements list)

Both kinds of traveling between workcenters (empty, full) are subject to the same "*pathing*" *decisions* (the combinations of track segments to be taken to get from the origin to destination) that are active at the time the AGV starts moving.

While *traveling*, an AGV might spend time in two ways:

1. moving: actually in action
2. waiting for track availability: the next track segment that it wants to take is currently occupied in the opposite direction: wait until it becomes available, allocate it in the desired direction and go.

After dropping a part at its destination workcenter, an AGV parks at the side track of that WC if there are no outstanding transportation requests queued.

It is obvious that any deterministic, static analytical model would be too naive and insufficient to capture the dynamic behavior of this Flexible Manufacturing System under the highly stochastic conditions mentioned above, such as random arrivals, variable part routing requirements, possibly random processing times, and dynamic AGV interactivity subject to instantaneously changing bi-directional track network availability.

A thorough simulation model is what we need to capture the behavior of this system realistically enough to be useful as a dynamic experimentational model and assist us in making AGV routing decisions. A detailed simulation model would help a great deal by forming a highly cost effective, risk free and accelerated (simulation clock can run faster than real time) testing environment for any kind of decision policies that we might want to try.

On the other hand, a merely *descriptive* simulation model does not ease our job to a great extent since the scenarios to be tried out are nearly inexhaustible (see the section named "All combinations considered"), and even the most experienced analyst can overlook some promising alternatives while designing test cases to simulate. The highly stochastic nature of the problem and variable part routing requirements with different arrival patterns (depending on changing demand/inventory states) make it impossible to bring the number of alternatives down to a reasonable few without carrying a non-trivial risk of losing those potentially promising alternatives.

AGV Routing:

The path alternatives from which AGVs can choose depend on the position of the destination workcenter with respect to the location of the origination workcenter. The number of paths are 5 for traveling to a neighboring workcenter (Figure 3), and 6 for a diagonal workcenter (Figure 4). Note that this is the default configuration considered, and any other dimensions in the solution space can be explored easily by changing the "paths input file" (a small portion of which is shown in Exhibit 2) that is read by the model to construct the necessary path and automata network configurations.

```
these_are_the_available_paths:
____track#_,_direction_940310

NumChoNeighbor
5
NumChoDiagonal
6

wc1_to_wc2__shortcut_[121]
1        +1
2        +1
0        0
wc1_to_wc2__short_u_turn_[122]
7        +1
3        +1
4        +1
9        -1
0        0
wc1_to_wc2__long_u_turn_[123]
......
```

Exhibit 2: the paths file

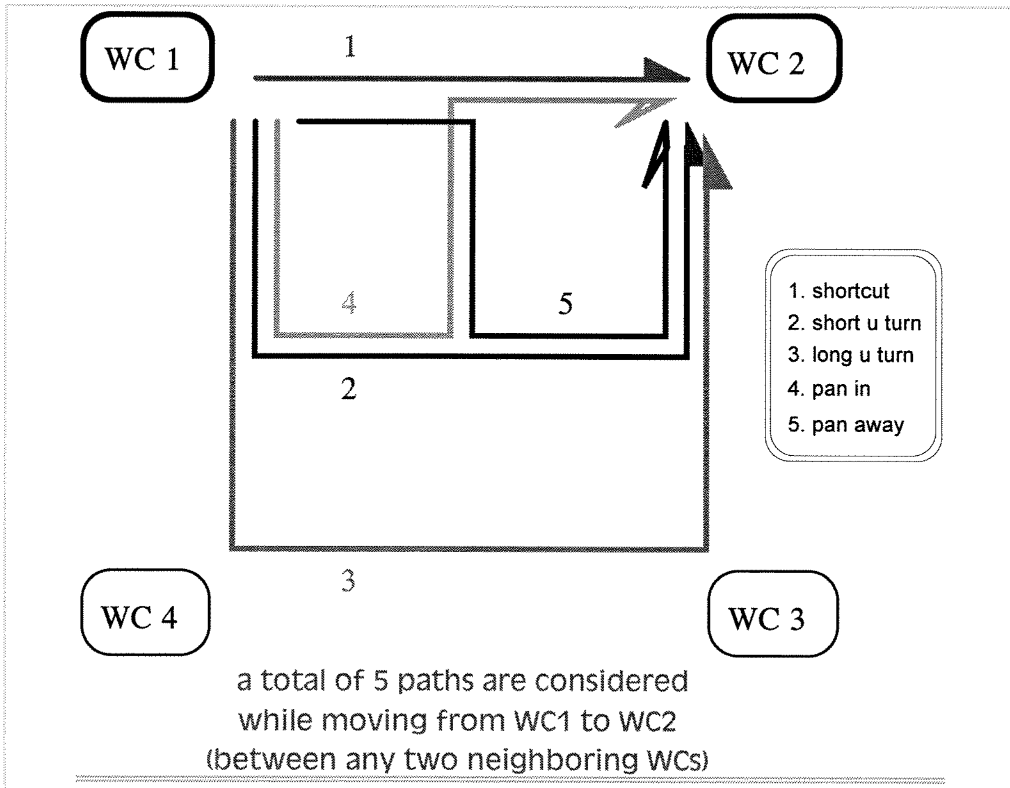


Figure 3: graphical representation of
Paths from WC#1 to WC#2 (neighbor)

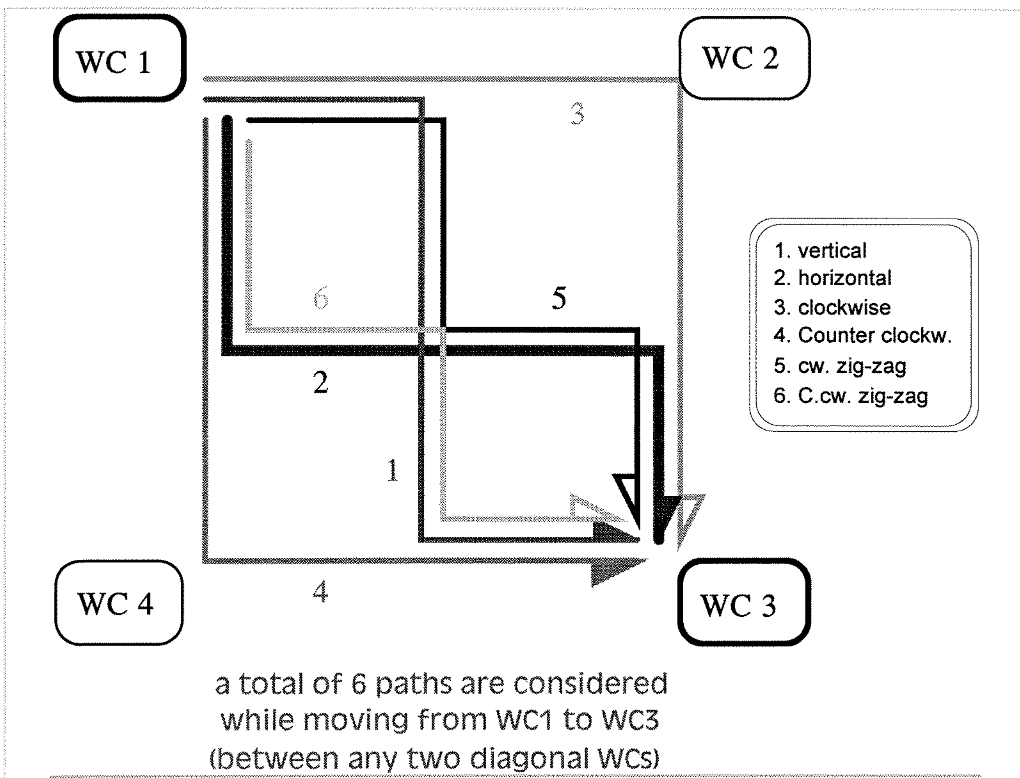


Figure 4: graphical representation of
Paths from WC#1 to WC#3 (diagonal)

All combinations considered:

The following calculations demonstrate the huge number of possible material handling system configuration scenarios to choose from: Since each workcenter has 2 neighboring WCs and 1 diagonal WC which have 5 (each) and 6 path choices respectively, there are a total of (5+5+6)*4=64 distinct paths available. An exhaustive enumeration of all possible decision policies with these paths will produce an almost *frightening* result:

At every workcenter, there are 5 alternatives to choose from to go to each neighboring WC, and 6 alternatives to choose from to travel to the diagonal WC. This makes 5*5*6 = 150 unique decision sets to choose from at every workcenter. Considering all four workcenters gives 150 raised to the fourth power, which is more than ***HALF A BILLION*** unique decision policy combinations to choose from...

Taking shortcuts is not always the best choice:

Whether we are routing empty AGVs to respond to part transportation requests or using AGVs to transport parts between workcenters, we have to make sure that the *summation* of net moving times (ie. directly proportional to the distance traveled) *and* **waiting times** (during which AGVs are blocked because the next track to be taken is not available in the desired direction) are minimized. Therefore, asking AGVs to take shortcuts wherever they are going is not always the best choice for overall fastest transportation of all parts.

Instead of wasting time by pulling off to the side track, waiting until the next track segment is available in the right direction and pulling back on the next track, it may be more desirable for the overall minimization of the generic part flow time to take a relatively longer path.

Waiting for availability of the next track segment could take a very long time if there is heavy traffic flowing in the opposite direction, for track segment allocation is *not* made on a FCFS basis: AGVs arriving at the other end of the track segment will be let in right away if the segment is available in their direction at the time thay arrive, even though they might have arrived *after* the AGV which is trying to grab the segment in the other direction. What is more, uncommon and inconsistent track allocations can potentially delay the possible heavy traffic trying to flow in the opposing direction.

The alternative (longer) path could prove to be more desirable even for that specific part type, as well as the overall prodiction rate: Although it will increase the net traveling time of that part, the potential decrease in waiting time may be more than enough to compensate for that time loss.

Variable, dynamic production schedule:

The FMS under consideration is capable of manufacturing virtually any part type that needs to be processed in the three workcenters (WC#s 2 through 4) with any permutation of routing requirements (including multiple visits to the same workcenter) and processing times. The idea behind this study is to deal with cases where the production of different part types will be dynamically started or discontinued. Even if the FMS keeps on manufacturing the exact same set of part types, any changes in the arrivals percent distribution (varying demand patterns and inventory status) may cause significant shifts in the optimal material handling system configuration.

An "intelligent simulation model" will allow the analysts to:

- ⋆ try out and see the performance of any suggested solutions (imposing the complete solution),
- ⋆ impose varying degrees of *partial solutions* to the problem: ruling out (or decreasing the probability of) some obviously inferior decisions, or imposing (or increasing the probability of) some definitely desired decisions; then letting the intelligent model experiment in the limited solution space with different exponential smoothing and learning constant values, and converge to a solution that is suggested by the quantified reward function; and finally
- ⋆ run the model without any restrictions, and let the model investigate the complete solution space on its own, again with different exponential smoothing and learning constant values.

II.B.3. Model Parameters

Simulation parameters are input at the beginning of every simulation run. Default values of entry fields in the dialog box shown in Figure 5 are populated by values read from an input file (Exhibit 3), and the user is given a chance to modify these defaults before starting the simulation run. Although it is possible to change some values throughout the simulation run, this is not recommended since it may have adverse effects on the learning process.

```
this_is_the_initial_
simulation_parameters_
input_file_for_CLEVER.exe


NumPalettes
10
NumAGVs
5
AGV_time_per_track_segment
1.0

NoOfWarmUpBatches
0
XAvgTISconst
0.01
GeneralLearningConst
0.01
DecnBatchSize
50

animation_speed_coefficient
1.0
XAvgConstStats
0.1

ProbabsAsBarCharts
TRUE
TISProbabsMOD
10
SimplifiedVersion
0
```

Exhibit 3

There are several values in the input file that do *not* appear in the initial simulation parameters dialog:

If "ProbabsAsBarCharts" is TRUE, then the dynamic interactive automaton control charts appear as *bar charts* that show the current snapshots of probability values, instead of *trace charts* which provide limited historical trend displays for the discrete distribution probabilities.

TISProbabsMOD is the plotting period (1/frequency) for the "exponentially averaged Part Times in System" trace chart that appears in "simulation control window", and the dynamic automaton trace charts (unless they are presented in the "bar chart" mode) that appear in the automaton control windows. This number is also used to determine how frequent part flow times and decision probability values will be logged to the output file.

If "SimplifiedVersion" is 1, the windows and other display/input objects for dynamic interactive automaton trace charts and dynamic monitoring of initial and exponentially averaged statistics are not created. Only the animation, simulation control and standard text I/O windows are displayed.

Number of palettes determines the maximum number of parts that can co-exist in the system at any time. The constant in line 5 (Figure 5) is used to exponentially smooth the trend in the flow time for different kinds of parts, including the generic part[4]. Line 6 (Figure 5) is the constant that is used in updating the automata probability distributions after the reward function is generated. The constant input at the last line (Figure 5) is used to exponentially smooth the trend in different state statistics collected about part types, workcenters, and AGVs.

In this study, a ***batch*** refers to a group of parts accumulated for an evaluation session by the automata network (please see section III.A for a detailed explanation of the *automata network*). The words *batch* and *decision batch* are used interchangeably. This quantity is used in two ways:

1. the loading/unloading workcenter ASKs the automata network to make new routing decisions based on their current probability distributions after *releasing* this many parts,
2. the automata network waits to collect this many parts before evaluating the batch and ASKing its automaton objects to update their probability distributions with the reward function generated from that batch.

---

[4] see section III.B.1 for the concept of a "generic part".

There are two points about releasing/collecting parts and generating/updating the automata that need clarification:

First of all, obviously, there will be a time lag between the end-of-release of a batch and the end-of-collection of a batch. Since the new set of decisions are made just before releasing the first part of the new batch (a little after end-of-release), and there still are parts in the system belonging to the previous batch at that time, the system goes through a transitory period (a non-steady state) between every batch. That means, some parts at the end of each batch will have to travel through the system while the decisions belonging to the next batch are being imposed on the system. These are called "**transitory parts.**"

The percentage of transitory parts and the overall effect of upcoming decision policies on the current batch can be reduced by:

1. Increasing the decision batch size: As a desirable side effect, this will increase the *reliability* of the feedback provided by each batch, and enable us to use greater learning constants with less risk of converging to wrong solutions. However, it can still slow down the overall learning process, since it reduces the frequency of updates made on the learning automata.
2. Reducing the number of palettes in the system: This is a parameter that will affect the operation of the FMS being modeled, hence it should not be modified arbitrarily to improve convergence of the learning algorithm.

The second point that needs to be clarified is that a few parts with smaller cycle times that were released at the beginning of the latter batch may exit the system (arrive at the automata network to be evaluated) *before* some parts with greater cycle times that were released at the end of the former batch. This further increases the "transitory period" that exists between the batches, and creates complications in batch evaluation.

After the user is satisfied with the values in the initial simulation parameters dialog box and clicks on "OK", a range checking for each value field is performed. If one or more of the values violate their valid ranges, the dialog box is popped up again with a warning. Otherwise, the program goes on to input the name of the file (Exhibit 4) that contains information regarding the number of different types of parts to be manufactured, their incoming percentage distributions, routing requirements (WC number in the first column), and respective processing times at each workcenter (second column).

All parts enter the system by being loaded on palettes at WC#1, travel through the system, and exit from WC#1 again, after being unloaded from the palettes.

```
this_is_parts_info._file_1

Number_of_types_of_parts=
2

part_type_1
.50
1       2.
2       5.
1       1.
0       0.

Part_type_2
.50
1       2.
4       5.
1       1.
0       0.
```

Exhibit 4

The last initial input to the model is the name of the results file that will be used to report detailed information about the simulation run. A summary of the input parameters and estimated lower bound for the flow time of the generic part will be output to this file before simulation starts. After completion of every TISProbabsMOD batches (the number representing the reporting period = frequency^{-1}), information about exponentially smoothed flow times for different kinds of parts, the average generic part flow time for the current batch, the exponentially smoothed generic part flow time, and the corresponding reward function generated are logged, along with the probability vectors of every automaton. After the simulation run is over, all statistics collected throughout the run is also dumped into this file for post-runtime analysis.

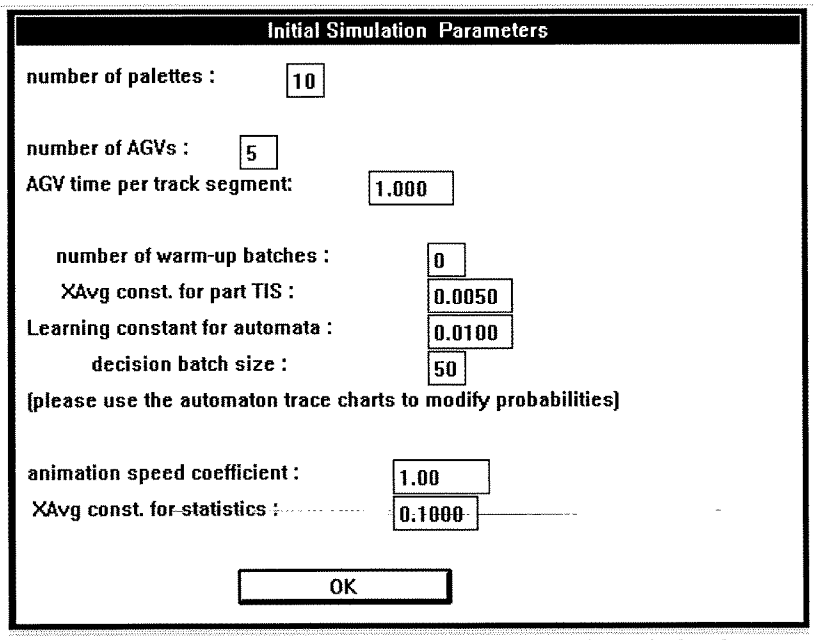


Figure 5: Snapshot of the initial parameter values dialog box.

II.B.4. Statistical Analysis

The following statistical information is collected during the simulation runtime:

- separate status statistics for each workcenter:
    - Time weighted *initial* statistics (WC States during the first 10 batches), and
    - Time weighted, *exponentially smoothed* statistics (more weight given to states during recent batches)

  for the following statistical entities:
    - percent time idle
    - percent time processing
    - percent time blocked
    - input buffer contents
    - output buffer contents

- aggregate AGV status statistics:
    - Time weighted *initial* statistics (states during the first 10 batches), and
    - Time weighted, *exponentially smoothed* statistics (more weight given to states during recent batches)

  for the percent time spent in the following states:
    - idle
    - FullMove: carrying a part to its next destination WC
    - EmptyMove: responding to an AGV request from another WC
    - WaitingMachine: the input buffer is full, cannot unload
    - WaitingTrack: next track to be taken is unavailable in the requested direction

- separate time consumption profile statistics for each part type manufactured:
    - Time weighted *initial* statistics (for the parts in the first 10 batches), and
    - Time weighted, *exponentially smoothed* statistics (more weight given to parts in recent batches)

  for the percentage time spent in the following part states:
    - processing
    - WaitForMach: in the input buffer, or on AGV waiting for input buffer
    - WaitForAGV: in the output buffer
    - AGVMove: net traveling time
    - AGVBlocked: being transported, but AGV is blocked (next track unavailable)

The initial and exponentially smoothed statistics collected about time consumption profiles of different part types, workcenters, and AGVs provide the user with an idea about utilization of the resources and the general conditions under which the FMS operates. These statistics serve as **secondary performance measures**: they *do* tell us about how the system is performing; but unlike the individual batch and exponentially smoothed Generic Batch Flow Time variables, they do *not* have a direct *closed-loop* effect on the reward function and hence the convergent learning behavior of the model. They are collected for monitoring and post-runtime analysis purposes only.

Please see Figures 6 through 8 for example snapshots of windows displaying these statistics.

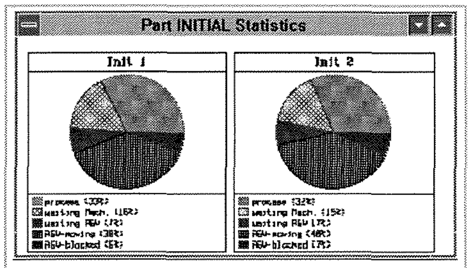


Figure 6: part initial statistics graphs

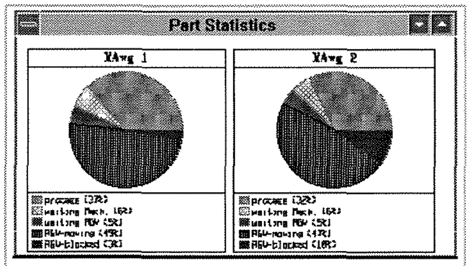


Figure 7: part XAvg statistics graphs

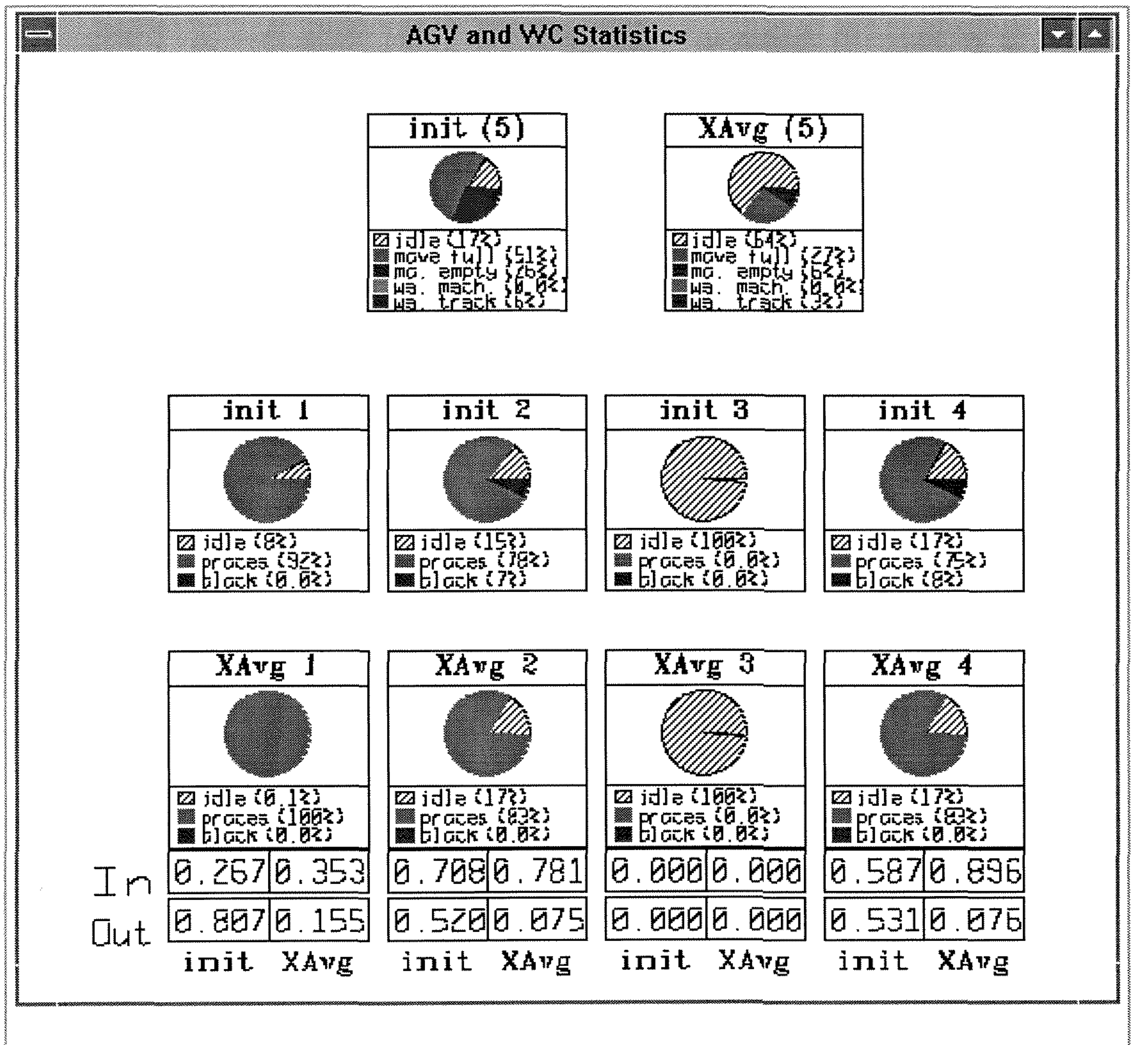


Figure 8: AGV and WC initial and exponentially smoothed statistics

III. EVOLUTIONARY DECISION MAKING

III.A. Topology of the Learning Automata Network

There are two types of learning automata utilized in this study. Both are updated by the same reward function (the reward-inaction learning algorithm summarized in Exhibit 1) and share the same learning constant, which is input to the model before simulation starts running. The difference is in their discrete probability distribution functions.

By default, the automata that control the AGVs going to a neighboring WC have 5 choices, and the automata in charge of sending AGVs across the room to diagonal WCs have 6 options to choose from (Figure 9). These defaults can be changed by modifying the paths declaration input file (Exhibit 4).

The automata that make decisions about AGVs which request to travel out of each workcenter are connected by WC switches. These switches select the appropriate automata according to the destination WC that the AGV wants to go (Figure 10).

There are a total of four switches, one corresponding to each workcenter, and each of those switches is in control of three automata that make routing decisions about outgoing AGVs (Figure 11).

Although the topology of the automata network suggests that decision making units act completely independent of each other, they do work in cooperation. This issue will be clarified in the next section.

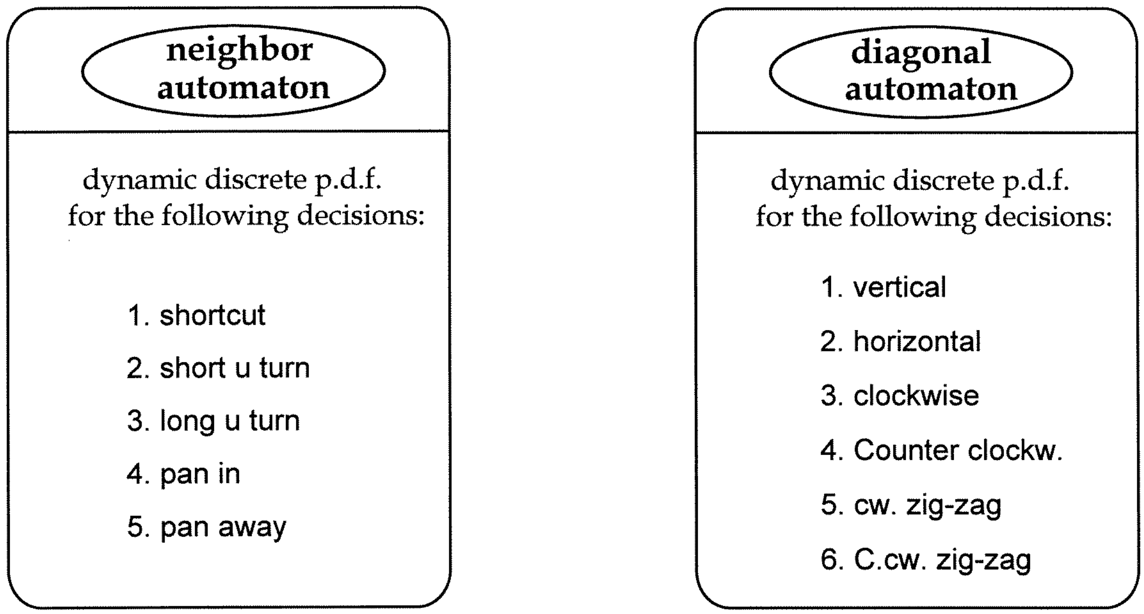


Figure 9: Probability density functions for generic neighbor and diagonal automata

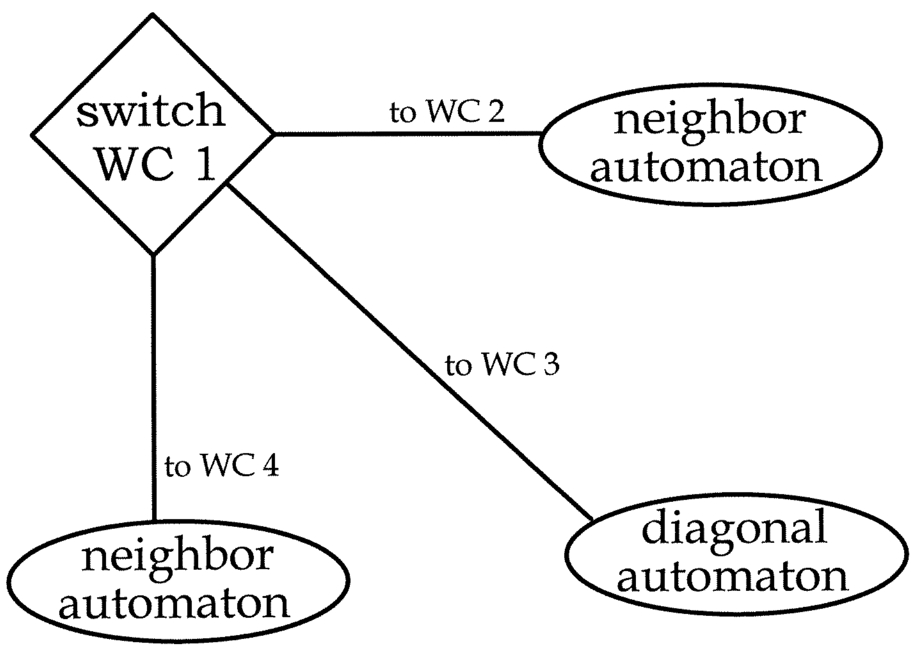


Figure 10: Workcenter switch #1 and its automata

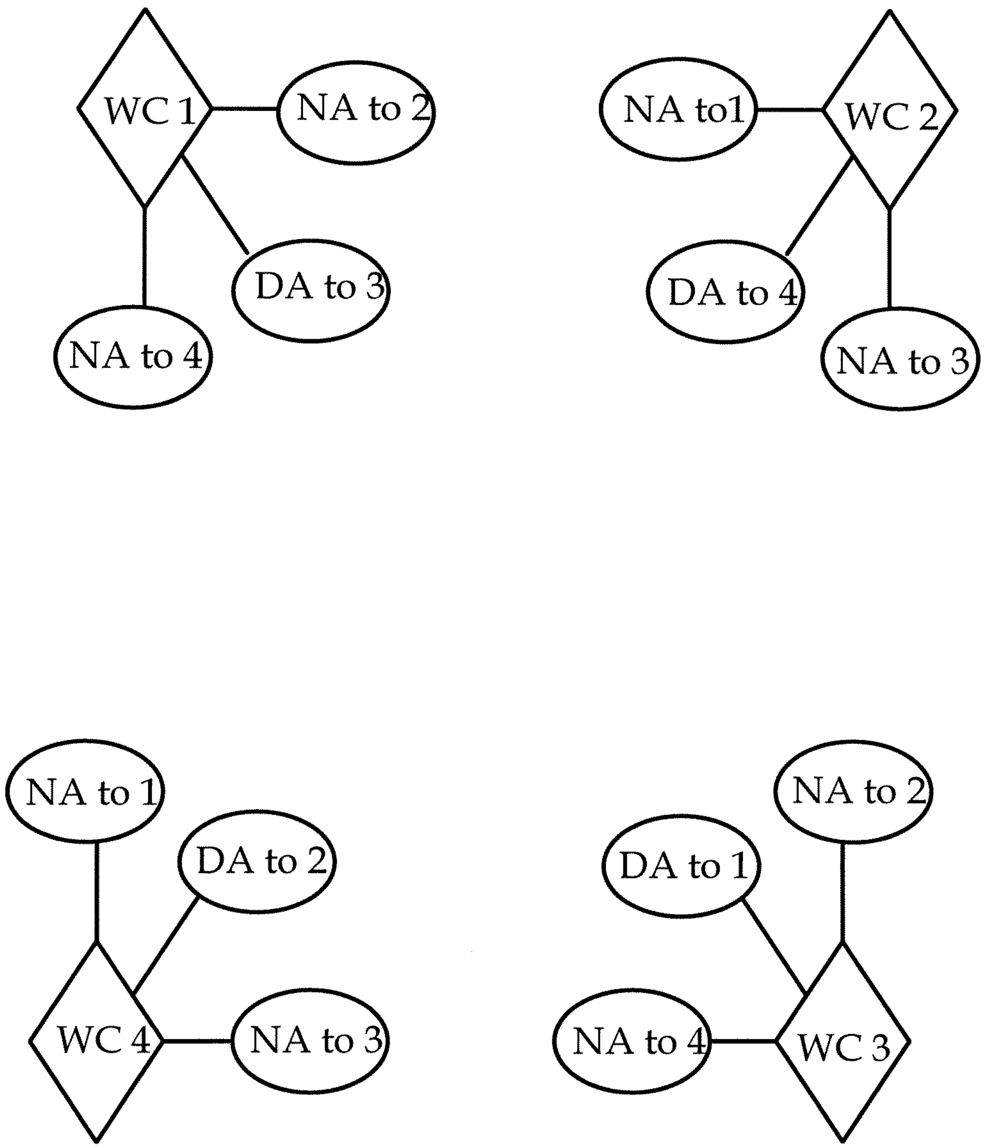


Figure 11: All four switches - the big picture
(NA = Neighbor Automaton , DA = Diagonal Automaton)

III.B. The Rewarding Scheme

The automata make decisions about the material handling system configuration (AGV routing decisions) right before each batch is started to be *released* (before releasing 1st, 51st, 101st parts, if the batch size is 50). The *same* system configuration is used until another batch is started.

The corresponding decisions of the participating automata are updated right after the loading/unloading workcenter completes *collecting* a "decision batch" of parts. Please see the model parameters (section II.B.3) for more detail on model parameters affecting the rewarding scheme.

III.B.1. the Reward Function

Flow Time:

The "flow" time for a part is defined as the total simulated time that elapses between the part's introduction to the system by the loading/unloading workcenter (WC#1) until the time that the part is unloaded from the palette and sent to the automata network.

A part is introduced to the system when WC#1 pulls it from the unlimited pool of different kinds of parts available (dynamically create the part, and decide about the type according to the percentage arrival random number distribution in the simulation context). After being introduced to the system, it undergoes its first operation: loading onto a palette. It then travels through the system on AGVs, visits every workcenter it has to visit depending on the routing requirements of its part type; and eventually returns back to WC#1. The part is unloaded, the palette freed, flow time calculated, and the part is sent to the automata network to be stored until the last part belonging to the same batch is collected and the batch is ready for evaluation.

the Generic Part:

In order to enable the conversion, comparison and standardized evaluation of different flow times for different types of parts concurrently being produced in the system, the concept of a *generic part* is introduced:

The generic part has a flow time equal to the weighted mean flow time of all different part types. The arrival percentage distribution is used to provide the weights for both the batch generic flow time average (Exhibit 5), and the *absolute* lower bound for the generic part flow time (Exhibit 6).

The absolute lower bound for the generic part flow time is used in calculating the reward function. It helps differentiate between similar relative reductions in the flow time by providing a more ralistic lower bound on the flow time than 0 and enabling the consideration of how close the system is to the optimal solution. It also helps avoid the decision makers get stuck in local optima.

Assume that the FMS is producing two different part types (1,2) with arrival percentages of 75% and 25% respectively. If the current batch reveals an *average* flow time of 40.0 time units for part type 1, and 60.0 for part type 2, then the generic mean flow time for this batch is calculated as follows:

$$(0.75) \cdot (40.0) + (0.25) \cdot (60.0) = 30.0 + 15.0 = 45.0 \text{ time units.}$$

Note that the size of the batch is *assumed* to be large enough so that the actual distribution of parts within instances of each batch would not deviate significantly from the theoretical distribution. The expected values of part ratios are used in the calculations instead of real ratios in the particular batch being evaluated.

Exhibit 5: mean generic flow time for a batch

Assume that the two part types manufactured in the FMS mentioned in Exhibit 5 have the following routing requirements (pairs represent WC#, processing time):

part type 1: {1,4 ; 2,15 ; 1,3}

part type 2: {1,4 ; 3,10 ; 4,5 ; 1,3}

Assume further that the AGV time per track segment is 1.0 time units, and the side track lengths for loading / unloading is 0.5 of the regular track segments.

The absolute lower bound for each part type is:

type 1: $4+.5\cdot1+2\cdot1+.5\cdot1+15+.5\cdot1+2\cdot1+.5\cdot1+3 = 28$

type 2: $4+.5\cdot1+4\cdot1+.5\cdot1+10+.5\cdot1+2\cdot1+.5\cdot1+$
$5+.5\cdot1+2\cdot1+.5\cdot1+3 = 33$

Note that these lower bound are based on the shortest possible path between workcenters and do not account for any delays incurred while parts are waiting in the input / output buffers or waiting for track availability.

The lower bound for the generic part time for this scenario is calculated as follows:

$$(0.75)\cdot(28) + (0.25)\cdot(33) = 21.0+8.25 = 29.25 \text{ time units.}$$

Exhibit 6

Exponential Smoothing:

Exponentially smoothed variables are kept up to date for the following quantities:

- the mean generic flow time, for comparative analyses while constructing the reward function of a new batch.
- all other (actual) parts' mean flow times, for monitoring and display purposes only.

See Exhibit 7 for a simple reminder about exponential smoothed averages.

If x_i is the i^{th} individual observation and $XAvg_i$ represents the value of the exponentially smoothed average after the i^{th} observation, then:

$$XAvg_i = (1-\alpha) \cdot XAvg_{i-1} + \alpha \cdot x_i$$

where

α = the *smoothing constant* in the range (0,1].

Exhibit 7

Calculating the Reward Function Value for a Batch:

To come up with an objective function for the overall system and a quantifiable performance measure for the individual batches to be evaluated, the concept of a generic part type is utilized:

When the average flow time for all the parts in a batch is calculated in terms of the generic part, the mean generic flow time of the whole batch is found (demonstrated in Exhibit 5).

The mean generic flow time is used as the performance measure for the set of decision policies that has controlled the AGV routings during that batch.

Assume that the exponentially smoothed generic flow time is 48.0 *before* the evaluation of the current batch for the FMS outlined in Exhibits 5 and 6.

If the current batch reveals a generic mean flow time of 45.0 time units {as demonstrated in Exhibit 5} and the absolute lower bound for the generic part type flow time is 29.25 time units {as in Exhibit 6}, then the reward function for that batch is calculated as follows:

β = current apparent improvement / absolute upper bound on improvement
= (48.0 - 45.0) / (48.0 - 29.25)
= 3.0 / 18.75
= 0.16

Exhibit 8: Sample reward function calculation.

To calculate the value of the reward function for a batch, the mean generic flow time of that batch is compared to the exponentially smoothed average generic flow time up to (but not including) that batch. If the comparison indicates that the performance measure for that batch was better than the exponentially smoothed value, then the decision policies that were active during that batch are updated with a positive reward function (β). Otherwise, the batch is ignored and the decision making probability distributions in the automata network are not updated at all (hence the "inaction" part of the reward-inaction learning algorithm).

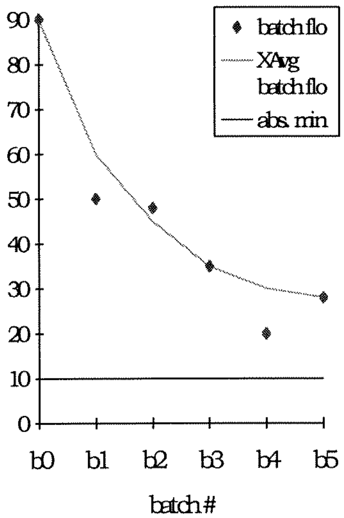
See Figure 12 for a graphical representation of the exponentially smoothed generic batch time, individual generic batch flow times, and more explanation on the the calculation of the reward function.

III.B.2. Rewarding the Automata

Once the value of the reward function is determined for a decision batch, all automata that have contributed decisions to routing AGVs in that batch have to be rewarded, proportional to the intensity of their involvement. The *intensity* of a decision during a batch is represented with the number of times it was used throughout the production of the batch.

Whether a decision was used to route an AGV out of the workcenter at which it had parked to respond to a workcenter that has an outstanding part routing request, or to route an AGV out of the requesting workcenter to take the loaded part to its next destination does not make a difference, since loaded and empty AGVs behave exactly the same in the track network traffic.

the reward function (β)



→no reward function is generated for batches b0,b2, b3 and b5; since the generic batch flow time is equal to or worse than the corresponding exponentially smoothed generic batch flow time

→although the absolute difference between current generic batch flow time and smoothed generic batch flow time are the same in both cases, b for batch b1 is less than b for batch b4, because the system is closer to the best theoretical state in the latter case.

$$\beta = \frac{\text{current apparent improvement}}{\text{absolute upper bound on improvement}}$$

$$= \frac{\max\left[0\ ,\ (\text{XAvgBatchFlow - CurrentBatchFlow})\right]}{(\text{XAvgBatchFlow - absolute lower bound on generic batch time})}$$

Figure 12: Graphical representation of exponentially smoothed and individual generic batch flow times and calculation of the reward function.

The automata are updated with the same reward function value calculated for the batch, as many times as their decisions were used during that batch. The decisions active while routing parts are logged onto the part objects during the time of routing, then they are retrieved and used during batch evaluation and update.

The decisions that were used to route empty AGVs have to be handled separately though, since for that kind of decisions, there are no semantic objects to log information onto. Instead, every empty AGV sends an "AGV brain record" to the automata network before taking off from the workcenter that they parked at. That record contains information about the active batch number, origination and destination workcenters and the then-current decision for that route. These records are stored in a queue similar to the one that stores the parts sent to the automata network, and kept in file until the batch is done being collected and ready to be evaluated.

III.B.3. Implicitly Coordinated Convergence

Despite the fact that decision makers are technically being rewarded separately, the response that they get back from the system *implicitly* includes the complex interactions that exist between the individual decisions made. A single *common* reward function is generated, reflecting the concurrent impacts of all decisions made about AGV routings during a batch, and no automata are rewarded if the individual decisions fail to work in harmony to produce an overall improvement with respect to the unifying objective function.

The result is a network of relatively independent, self-contained decision making units with an "unselfish" rewarding scheme that avoids local, individual convergence patterns ignorant of others; and provides cooperative, coordinated convergence for overall system improvement.

III.C. Dynamic Monitoring and Interactive Modeling

There are four dynamic interactive monitoring windows, one for each automata group that cluster around one of the four workcenters. They provide *dynamic*, ongoing status updates for the internal decision making probability distributions, and the current decisions of each of the three automata in every group. They are *interactive*, since the user can click on any one of the dynamic charts at anytime during the simulation run, not only to see the precise values, but even *modify* the internal decision making probability density function of any of the learning automata.

The dynamic interactive automaton control charts can assume two forms, depending on the input provided in the initial simulation parameters file (see Exhibit 2): either a *bar chart* (Figure 13) that shows the snapshots of current probability values, or a *trace chart* (Figure 14) which provides limited historical trend displays for the probability values.

There is another very important object that provides means of supporting user interactivity throughout the simulation runtime and promotes the visually interactive modeling (VIM) environment: The menu-bar present in the simulation control window (Figure 15) acts as a means of interrupting the simulation run at any time, and give the user the flexibility to modify any parameters whenever necessary.

The menu items are mainly used to allow the user to modify animation speed, switch between the animated mode and relatively faster running no-animation mode, and let the user dynamically change the values of some other parameters that were input before beginning the simulation run. The default organization of the menu items are shown in Figures 16 through 19, but not all of them are implemented in the current version.

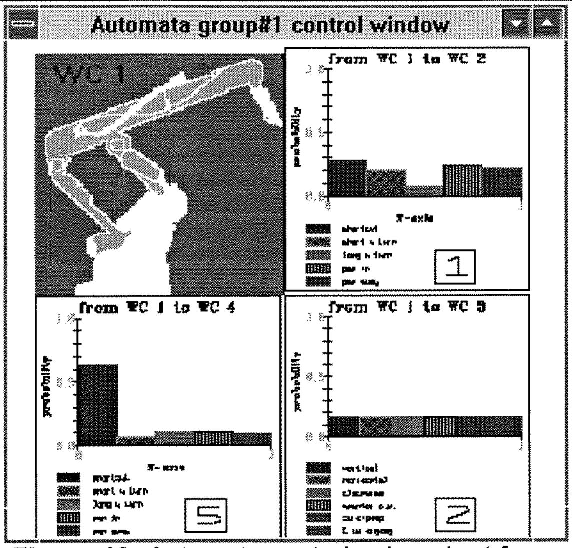


Figure 13: Automata controls - bar chart form.

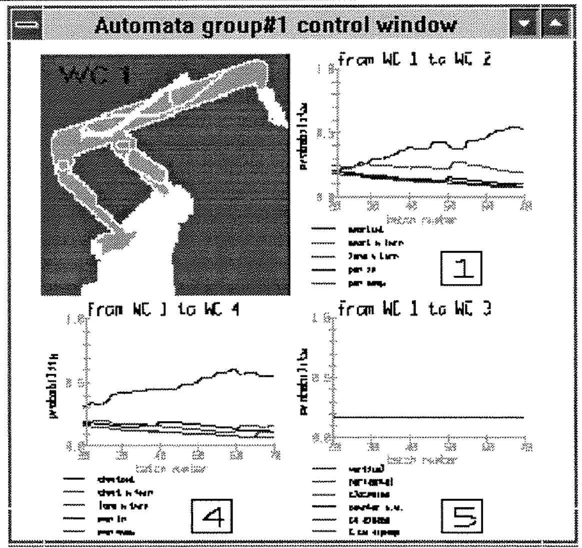


Figure 14: Automata controls - trace chart form.

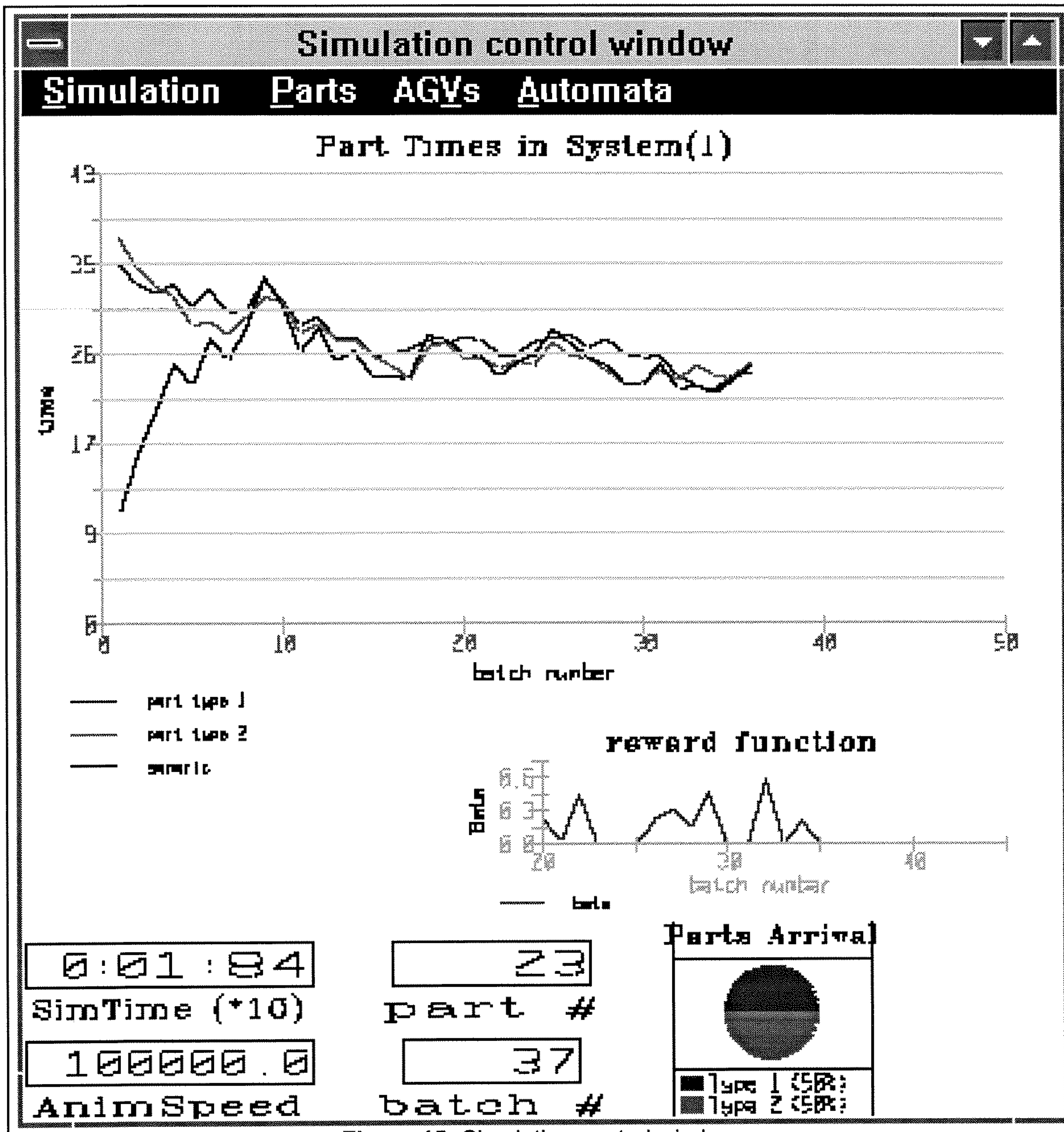


Figure 15: Simulation control window.

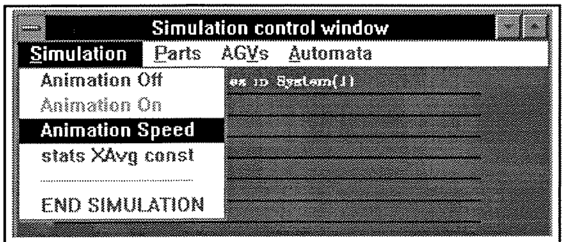


Figure 16: Simulation menu.

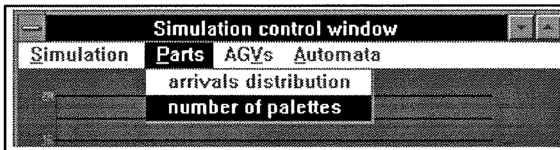


Figure 17: .Parts menu.

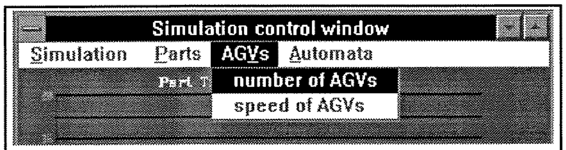


Figure 18: AGVs menu.

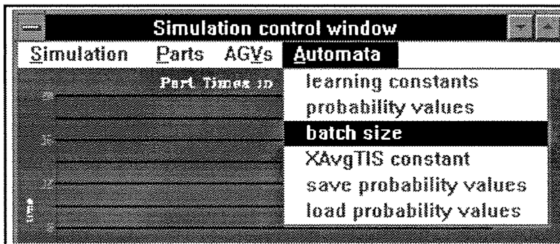


Figure 19: Automata menu.

IV. IMPLEMENTATION

IV.A. Classification of Choices

The dynamic relationships between modeled entities imposes the time elapsing mechanism to be used, and divides models into two distinctly differentiable sets, namely Discrete Event Simulation and Continuous Simulation. This study is concerned with the former, where state transitions occur at discrete points in time.

IV.A.1. Implementation Platform

General Purpose Programing Languages:

There are many examples of simulation models implemented using conventional programming languages like FORTRAN, BASIC, Pascal, C or C++. Unfortunately, in almost all of these examples, the majority of the effort is spent in order to develop the infrastructure that simulation languages readily provide the user.

Some constructs like random number streams and distributions, statistical output collectors, resources and queues have to be built from scratch, because almost none of the general purpose programming languages provide these most basic building blocks common to all simulation models.

Elapsing simulation time and taking care of multiple processes at the same instant with respect to the simulation time is another concept crucial to simulation modeling that does not go together naturally with general purpose programming languages: A global variable representing simulation time, and a Future Events List (FEL) representing the upcoming interactions among components in the model has to be kept up to date. These events have to be scheduled, initiated, and disposed of; explicitly by the programmer. This process can be extremely difficult and even the most thoroughly planned programs will be limited to relatively small scale simulation models, especially if the general purpose programming language does not allow the use of dynamic data structures.

Design, implementation, testing and debugging becomes extremely difficult in this "forced", unnatural environment. Not only the simulation model and the concepts abstracted from the real system under study, but the infrastructure formed using the general purpose programming language has to be passed through rigorous verification and validation processes.

A strong point of simulation is that it allows the analyst to incorporate randomness that exists in the real system into the computer model. There can be complex, dynamic relationships between these stochastic components of the model, which will cause further complications in a general purpose language implementation. For example, a *preemptive* resource downtime can interrupt, and effectively extend the processing duration of a part in a location. On the other side, the event record representing that part's leaving that location would have already been logged into the FEL at the time the service had started with a *static* random time tag, and it would not be valid any more, so it has to be updated after the preemptive downtime has occurred.

As a result, except for special cases which do not lend themselves to a natural *and feasible* solution with available simulation languages; and research projects whose sole aim is to explore the possibility of simulating with general purpose programming languages, I believe that simulation modeling should be implemented with languages especially developed for that purpose.

Simulation Languages:

Using simulation languages reduces the development time, and removes the risk involved with building the simulation infrastructure from scratch. Basic model building blocks like entities, resources, queues, random number generators with numerous distributions, run-time data collection and even statistical output analysis is provided as built in facilities.

Most importantly, the notion of passing simulation time is supported naturally in the language, so the highly error-prone process of scheduling events (and the burden of handling artificial data structures for this purpose) is removed from the model builders' shoulders.

As a result, the analysts can concentrate on the concept of modeling the system instead of worrying about explicitly developing the mechanisms which will support a simulation environment.

Some users have a fear of loss of generality and flexibility when simulation specific languages are used. However, most of today's simulation packages allow making calls to external routines when the flexibility of the original language is not sufficient. There are also a number of simulation languages (eg. MODSIM II) which are *almost* as powerful as the most competent general purpose languages like C++.

IV.A.2. Software Development Paradigm

The gap between conceptual solution domain and the implementation of those conceptual models in computer programming languages is very much dependent on the development paradigm used.

Procedural Programming:

Most simulation applications that are implemented with conventional languages up to now make use of the procedural (functional) paradigm of programming. After all, that is the overall most dominant programming paradigm in the world for any kind of computer programs.

Most simulation languages support variations of this paradigm, if they support any modularity at all:

SIMAN, for example, allows the model builder to make calls to user written subroutines (FORTRAN and possibly other languages depending on the platform) to handle the complex situations in which standard building blocks are not sufficient to fully represent the behavior of the real system. However, while executing the user code, the rest of the model (any operations on entities other than the one that has activated the user routine) halts: simulation time elapsing is not permitted in user routines.

ProModel for Windows, on the other hand, comes with the flexibility of being able to define time elapsing subroutines within the context of the original simulation language, which can be invoked from anywhere in the model (initialization, termination, operation, downtime and move logics, and even from within other subroutines).

Modularity helps organize the thought process and therefore aids model design and implementation phases. However, the procedural paradigm is **not** the most natural way of mapping from the problem domain to the computerized solution domain, especially for the case of simulation modeling. "The procedural paradigm takes a *task oriented* point of view[5]", as opposed to the object oriented paradigm which takes a *modeling point of view*, which is exactly what a simulation analysis is for (more about this idea in the next section).

---

[5] Korson, 1990.

The conceptual model and the resulting code will inevitably contain unnaturalities when the procedural paradigm is utilized: Elements will be bent out of shape while transforming from the real system to the solution domain and then to model implementation. The code will be extremely hard to test, debug and extend as the scope of the study expands, and it will be almost impossible to build more detail if the depth and accuracy of the model is to be increased.

Object Oriented Programming:

Simulation is a very appropriate context to make use of object oriented analysis, design and implementation. The object oriented paradigm fits more *naturally* while making the transformation from the real world system to the computerized simulation model, which means that the **semantic gap** between the problem and programming solution will be much smaller.

"It [the object oriented paradigm] is natural in the sense that the design pieces are closely identified with the real world concepts which they model. It is flexible in the sense of quickly adapting to changes in the problem specifications.[6]". Object oriented design and programming allows the development of highly reusable, easily modifiable, generic code. Utilizing the object oriented paradigm seems to be a must for obtaining a truly extensible general simulation infrastructure library, because of its inherent features like data hiding, encapsulation and inheritance.

Computer specialists may relatively easily see the data structures to represent the system and develop a functional decomposition of the processes going on: they are the computerized solution domain experts. On the other hand, especially in simulation analysis, experts in the real system (who probably do not have much experience in software development) have to get much more closely involved in the problem formulation and solution processes when compared to conventional programming projects.

These real life system experts usually think directly in terms of the problem domain entities: **objects**, instead of the *unnatural* procedures and data structures that result from the translation to the solution domain using a functional programming paradigm. The object oriented paradigm will do a much better job of tying together the real life state (attributes) and behavior (methods) of objects (like machines, workers, parts, bank tellers, customers, conveyors, etc.) with their computer model representations.

The physical objects and their relationships in the real system (hierarchical organization, interactions and behaviors) will have direct counterparts in the conceptual model and the computerized solution domain (through the use of inheritance, message passing, etc.). These objects, which exist both in the problem domain and the solution domain, are called **semantic objects**[7]. The semantic objects and their interactions will be easily, naturally transformed into the solution domain with the help of the object oriented paradigm.

---

6 Korson, 1990; p.41.

7 A definition from Monarchi & Puhr, 1992: "Problem-domain objects represent things or concepts used in describing the problem (rather than the solution). We call them semantic objects because they have meaning in the problem domain."

IV.A.3. Simulation Modeling World View

Most of the simulation languages will impose the usage of a specific world view on the user. If a general purpose programming language is used to build the simulation infrastructure, on the other hand, it is the programmer's choice which point of view to take.

Event Oriented:

The event oriented world view models a system by

- determining the events which may change the state of the system
- defining the logic that will be executed when those events happen
- defining the mechanism to schedule events on the time axis

In the pure event oriented approach, no time advance is possible within the event logic statements. Time advancing is handled by scheduling other events at future times within executed event logics.

Process Oriented:

Rather than focusing on the events that may change system state and defining logics to be executed at those discrete "turning points" in time; the process oriented approach tracks the progress of elements throughout their life cycles, defining the processes that they go through from the time they are introduced to the simulated system until they exit the system. [8]

In the relatively narrow[9] sense, "it [the process oriented world view] provides a description of the flow of the entities through a process consisting of resources.[10]" The entities can be parts, customers, or patients, and the resources can be machines, workers, bank tellers, nurses, doctors, etc. The simulation language translates the process oriented description of the system to a sequence of events, and handles the scheduling automatically.

The synergistic benefits that can be obtained by combining the process oriented world view and object oriented simulation languages will be mentioned in the following section.

---

[8] see Hoover & Perry, 1990; p. 100-106 for comparative examples including flowcharts, for both world views
[9] see section IV.B.2 for a broader meaning that arises from using the OO paradigm
[10] Pritsker, 1984.

IV.B. Best Choices

There is a big semantic gap between the conventional programming paradigm and the simulation concept. The solutions implemented using non-simulation languages are unnatural and significantly more difficult to debug, maintain and especially further extend. Investment made in simulation packages almost always pays off, both in business and academic environments. The question, then, boils down to choosing the most appropriate simulation language.

IV.B.1. Practical Issues

Simulation modelers are not limited to software development professionals any more. Engineers, managers and other experts about the real life systems to be explored are the ones who are increasingly taking a hands-on attitude to analysis and decision making using computers.

Therefore, it makes perfect sense to expect the development and usage of sophisticated simulation analysis environments with a simple front end interface, making the user independent of the internal complications of the environment.

A lot of easy to use, event or transaction flow oriented simulation packages are available in the market today. A big majority of them support the functional decomposition paradigm (which is quite familiar to even the most novice computer programmers) to bring more flexibility above the default structure of the simulation language. An increasing number utilize visual aids in model building and allow easy animation during simulation run-time (which is an important concept that lies outside the scope of this paper).

ProModel for Windows is a good example : It takes a hybrid world view, which mainly focuses on the flow of entities in the model; from their "arrival" (creation and introduction to the system) to their "exit" (going out of the system boundaries). The backbone of the simulation model is formed by a series of operation logic blocks, one corresponding to every "entity-x at location-y" combination, supported by various other logics including entity routing, resource entry/exit, etc. The behavior of the system is mainly represented by the event records which get executed when the entity arrives at the location, but a series of time elapsing statements can be inserted in the operation blocks, also.

ProModel eases model building and facilitates communication by making excellent use of the graphical user interface and animation. It seems to be an ideal tool for non-software development specialists to get quick results without spending excessive effort. However, as models get more and more demanding, the unnaturalities of the classical approach to modeling start showing up, and one ends up struggling with numerous process records with complex routing references driving the model in a highly disorganized way.

Taking the above advantages into consideration, the initial implementation platform for this study was chosen to be ProModel for Windows. See Figure 20 for the layout of the fairly detailed model constructed.

Later, it was decided that MODSIM II would be a more suitable development environment for the final version, given the exploding model size and complexity and the non-conventional features required by the nature of the study.

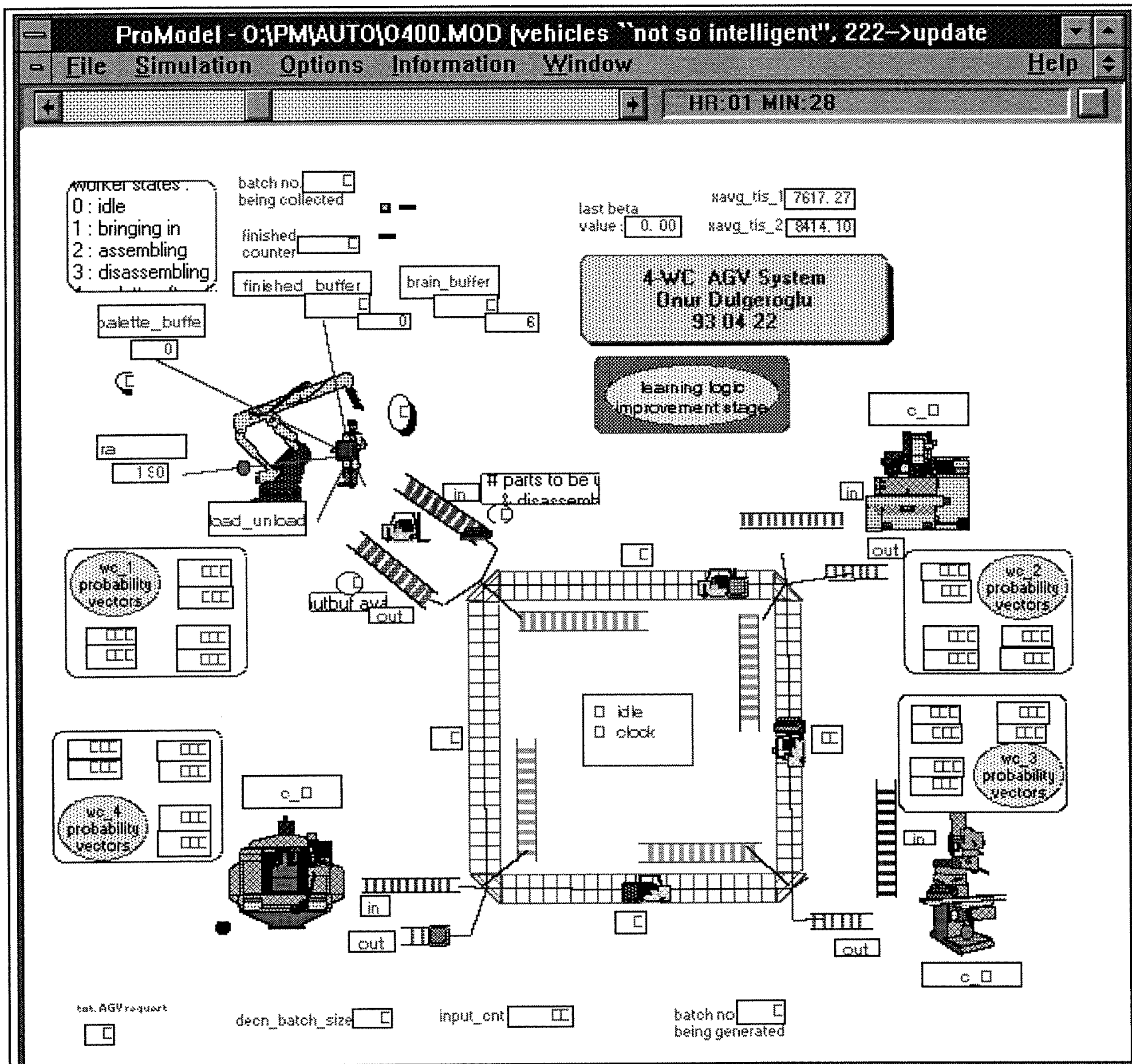


Figure 20: ProModel screen snapshot.

IV.B.2. the Best Combination

Why use an Object Oriented Simulation Language with a Process Oriented World View?

As the scope of the model enlarges to include more components from the real system and the level of detail that those components are to be represented increases, the complexity of the conceptual model and especially the computerized simulation implementation grows exponentially.

If the perspective taken to represent the system as a computer model is the event oriented world view, lots of event routines end up being written. Looking from the software development point of view, these events are literally unrelated, stand-alone routines in the computerized model. However, they actually represent closely related groups of behaviors in the real system and the conceptual model.

The logical flow of actions that entities go through, or the characteristic behavior of resources are scattered throughout the computer program instead of being integrated into more meaningful "packs". As the model gets more involved, it becomes more and more difficult even to follow the elements and the interactions between them, let alone modifying or extending those interactions. The computerized simulation model quickly assumes a form which is similar to a piece of "spaghetti code" with a lot of GOTO statements in the conventional programming analogy.

These complications are of extreme importance, especially if the simulation model being built is an operational one, which will run -and most probably **continually evolve**- over an extended period of time, possibly by different modelers and system experts to aid different decision makers with varying viewpoints on the system.

Using OOP may bring the "high *potential energy difference*" between the problem (the system, the concept of simulation) and the solution (the computerized model) domains down to a more stable level, but still: the concept of "*time elapsing routines*", the ability of *asynchronous message passing*, pseudo *multi-tasking* (multiple processes running at the same simulation instant), *invoking multiple methods of the same object at the same time*, or *running multiple copies of the same time-elapsing method* of the same object at the same time are not naturally possible in most platforms, and it may easily get out of control when emulated artificially, even in relatively small sized simulation models. The analyst / programmer will still have to spend considerable effort towards developing the simulation environment infrastructure and the emulation of concurrency.

For the analysts to be able to concentrate on the modeling and decision making process, an object oriented programming language which supports the concepts of passing simulation time and concurrency in method invocation would be the best (for instance, MODSIM II, an object oriented simulation language, supports various types of method invocations : namely ASK, TELL and WAITFOR methods).

Before the development of object oriented simulation languages, it would have been correct to say that "since we are normally restricted to a set of standardized statements provided by the simulation language, our modeling flexibility [with the process oriented world view] is not as great as with the event orientation.[11]" Today, if the implementation platform for the model is chosen to be an object oriented simulation language, the modeler can make use of the standard simulation objects provided in the language's object libraries, and can also extend the standard class structure vertically or horizontally, making use of inheritance to develop custom made entity, resource, or hybrid objects with enriched states and behaviors.

[11] Pritsker, 1984.

If the process oriented world view is approached with an object oriented design and programming paradigm, a much broader meaning may be associated with the *entities* and *resources* mentioned in section IV.A.3 (Process Oriented). Every element of the simulated system will be represented as objects, and they will have their own attributes (memory or state) and their behaviors (methods). Therefore, the modeler will not be limited to seeing the world from the point of view of some data structures called "entities" that flow through some static model building blocks called "resources".

"The object oriented design paradigm takes a modeling point of view. The analysis and design phases of the traditional life cycle ... work together closely to develop a model of the problem domain. The model is constructed by viewing the problem domain as a set of interacting *entities* ("entity" in a wider meaning than usually used in the simulation context). Software based models of the entities and relationships between them are assembled to form the basic architecture of the application. The information developed in the analysis phase becomes an integral part of design (and hence the implementation of the simulation model) rather than simply providing input to the phase.[12]"

In [Popken, 1992], the author separates the simulation modeling process into two distinct phases:

1. forming the conceptual model, which involves the "*essential* modeling complexity", and
2. translating the conceptual model into a computerized implementation, which brings up the issue of "*accidental* modeling complexity".

He adds that the complexities arising in the first stage are unavoidable by the very nature of the system that is being modeled, while object oriented programming helps avoid the complexities in the second phase by reducing the [semantic] gap between conceptual models and implementation of those models in computer programming languages.

According to Monarchi and Puhr (1992), object oriented design (OOD) methodologies assume that relevant problem domain objects have been identified. Through OOD, the details of each object's behavior and the inter-object communications (messages) necessary to meet system requirements are modeled. There is no doubt that OO design and implementation *will* contribute to higher quality computerized models developed from conceptual solutions, and therefore help reduce the *accidental modeling complexity* mentioned in stage 2 above.

What is more, Monarchi and Puhr (1992) also *imply* that the OO paradigm can help simulation analysts deal with the complexities that arise in the first stage (forming the conceptual model), as well. The authors define OO Analysis methods as "being concerned with abstracting relevant objects from the problem domain, defining the objects' structure and behavior, and determining object interrelationships", and the goal of OOA as being "to model the semantics of the problem, in terms of distinct but related objects"; which is the *exact tool* to attack the first phase of simulation analysis and keep the "*essential* modeling complexity" under control.

An object oriented solution to managing the complexity of a large design is also offered in Monarchi and Puhr (1992). It consists of defining various "abstraction layers" of the system, which directly applies to OO simulation modeling: The top, or macro layer(s) should communicate the system structure, function and control, and would be developed during conceptual model development. The bottom, or micro layers should represent the objects' structure, functionality and control, and would be developed during system analysis and/or object design.

Initially, an experimental prototype model can be built, representing the basic backbone of the system under study. This model may capture the most important objects in the problem domain, their "rough" behaviors, and the relationships between them. An initial inheritance hierarchy and public interface of the objects to be modeled can be constructed.

Later, more detail can be added on as needed, with ongoing validation, verification and data analysis. Roughly modeled objects can be further specified either by enhancing the base class definitions or by vertically extending the inheritance hierarchy and making use of derived, specialized classes.

---

[12] Korson, 1990, p.46.

IV.B.3. Examples of the Reduced Semantic Gap

MODSIM II is an object oriented simulation language with a process oriented world view. Except for some *fatal* bugs still present in version 1.9, it can be considered a sophisticated language due to the uniqueness of its semantics. It enjoys most of the benefits mentioned in previous sections, like allowing visual interactive model building, graphical animation, invoking multiple methods of the same object at the same time, or even have multiple copies of the same time-elapsing method of the same object running simultaneously.

MODSIM allows for three distinct kinds of method calls to objects:

- <u>ASK methods</u> are similar to C++ method calls. The called routine is executed immediately, and the caller routine halts execution until control returns to the line after the one from which the call was made.
- <u>TELL methods</u> make a very useful and unique class of method calls possible: TELL methods can elapse simulation time. The uniqueness arises from the fact that the caller routine *goes on with its execution* right after making the call to the TELL method. It does not wait for the invoked TELL method to be completed. What is more, caller routines can schedule future invocations to TELL methods by making use of the "IN" qualifier, as in the following example, calling for an AGV from a WorkCenter object:
  TELL FMS.AGVallocator TO SendAGVto(SELF) IN 0.1{simulation time units} ;
- <u>WAITFOR methods</u> are like ASK methods except that they can elapse simulation time. The caller routine halts execution after invoking, and goes on with the next statement only after the time elapsing WAITFOR method is done. ASK methods cannot make calls to WAITFOR methods, since they are not allowed to elapse simulation time.

The simulation power that these different kinds of method calls give the user, together with all the benefits of an object oriented language helps reduce the semantic gap between the real system to be modeled. The result is a very stable design and implementation for most complex cases. Some example method calls from MODSIM definition modules are displayed in Exhibit 9, and several complete methods taken from implementation modules are included in appendix D.

minimal semantic gap

invoking object invoked method type	invoking method invoked object name/type	optional	invoked method	parameter(s) passed
AGVallocator	SendAGVto			
TELL	ClosestAGV	TO	Serve	(RequestingWC)
AGV object	Serve			
WAIT FOR	SELF	TO	GotoDestination	
ASK	CustomerWC	TO	Give	(thePart)
ASK	thePart		NextDestination	
TELL	SELF	TO	MoveToSideTrackWC	
WAIT FOR	DestinationWC	TO	Take	(thePart)
WC object	KeepBusy			
WAIT FOR	InputBuffer	TO	ReceivePart	
ASK	InputBuffer	TO	Remove	(thePart)
WAIT FOR	thePart	TO	BeProcessed	
WAIT FOR	OutputBuffer	TO	AddPart	(thePart)

Exhibit 9: Examples of MODSIM II method calls

V. SAMPLE RUNS

CACI's MODSIM II version 1.9 has some bugs, which cause a big burden to development and debugging efforts. The executable code of models built using MODSIM are not always 100% windows compatible: they occasionally abnormally terminate themselves for no apparent reason.

What is more, CACI has recently admitted that there also is a fatal memory allocation/de-allocation problem: some objects and records automatically generated by the simulation engine (internal variables) are not disposed of properly after being used. As a result, the executable model compiled by MODSIM keeps *expanding* in memory until it fills all available RAM, and the Windows operating system starts swapping the inflated memory with a big file on the hard disk, used as virtual memory. After some time, the swapping becomes so intense and chaotic that none of the applications including the model that is being run can get their share of the CPU, and the model practically *halts execution.*

As a result of this severe "memory leakage", experimentations with the "intelligent" model built was limited to extremely simple cases, to be able to demonstrate convergence in a very short amount of time.

V.A. Run 1: Two Different Part Types

The inputs for the first case study are given in Exhibit 10. The FMS is run with 10 palettes and 5 AGVs, to manufacture the parts loaded from the parts file in Exhibit 11.

this is the clever simulation model - 940410	
system inputs :	
Number of palettes =	10
Number of AGVs =	5
AGV speed =	6553.4
AGV time per track segment =	1
number of warmup batches =	0
XAvgTISconst =	0.01
GeneralLearningConst =	0.01
Decision Batch Size =	50
XAvgConstStats =	0.1
parts file name = clever.par	
report file (THIS FILE) name =	res\clever.out

Exhibit 10: Run1 - input parameters.

parts file comment :: this_is_the_parts_info._file_1		
number of different types of parts = 2		
part type # 1 comment = part_type_1		
Percent arrival = 0.500000		
Process records : (WC# ; time)		
	1	2
	2	5
	1	1
part type # 2 comment = Part_type_2		
Percent arrival = 0.500000		
Process records : (WC# ; time)		
	1	2
	4	5
	1	1

Exhibit 11: Run 1 - parts file.

As can be seen from the trace charts of workcenter automata shown in Figure 21, the intelligent model converges to using shortcuts to go from WC#1 to both WC#2 and WC#4. Figures 22 and 23 show that the decisions converged to using shortcuts while coming back to WC#1 from both WC#2 and WC#4, as well.

Automata responsible for routing AGVs between WC#2 and WC#4 are *rarely* used to send empty AGVs from one workcenter to another to carry parts back to WC#1. As a result, they are not updated as frequently as others, and do not converge to specific decisions during the extent of this runtime.

Note that none of the parts need to be processed in workcenter 3, therefore no AGVs travel to that WC. As a result, the decisions are not used, and never updated (similarly, all the decision p.d.f.'s clustered around WC#3 stay untouched as a uniform distribution).

Figure 24 shows the exponentially smoothed flow times for both of the part types, as well as the generic part. Figure 25 demonstrates the decreasing trend in the exponentially smoothed generic flow time, and individual batch flow time values "dancing around" their exponential average. The thick line in Figure 26 represents the individual generic batch flow times that are exponentially smoothed during post-runtime analysis. The downward trend in this line is a solid proof of improvement resulting from the learning process that the automata are going through.

The general trend in percentage profile of flow times for each part type can be deducted by comparing the initial (first ten batches) and exponentially smoothed (all batches, more weight given to most recent ones) statistics collected about these part types. Multiplying these time profile distributions by *estimates* of flow time at corresponding batches, one can see the trend in the absolute time spent by every part type in each state. This analysis will not produce exact results, though: There will be some discrepancies due to the fact that the two sets of values used (namely, flow times and time profile distributions) are *not* collected in a fully compatible way. The obvious source of this incompatibility is the use of different smoothing constants while calculating exponential averages. Also, for the initial set of statictics, percentage time profiles are calculated by giving equal weights to the first ten batches, whereas flow times are always calculated by exponetially smoothing individual batch values.

Figures 27 and 28 demonstrate the results for this approximate analysis for the two parts: It can be observed that as the learning automata converge to better solutions, the overall time that both types of parts spend in the system decrease (overall decrease in the cumulative bar heights for all categories) and parts start spending much less time in the output buffers of workcenters to be picked up by the AGVs, and less time actually being transported on AGVs (due to the decrease in the net length of paths taken by AGVs).

The fact that these graphs show different net processing times for initial and exponentially smoothed statistics is due to the discrepancy mentioned above. The effects of this problem will be reduced considerably, and statistics will be more heathy in the second case study in which only a single part type is manufactured.

Similar analyses involving the comparison of initial and exponentially smoothed statistics are shown in Figures 29 through 32 for aggregate AGV profiles and workcenter profiles for WC#s 1, 2 and 4 respectively. The difference between the part statistics analyses and these is that these show the trend in the *percentage* time profiles of the respective resources, whereas the part analyses show the percentage profile multiplied with the average flow time of the respective parts. There is a decrease in the cumulative bar heights in Figures 27 and 28 (parts analyses) due to the decreasing trend in flow time, but the bar heights in Figures 29 through 32 will always sum up to 100%, reflecting the change in percent time spent during different activities as the automata converge to better decisions.

Figure 29 indicates that the AGVs have more idle time and that the system could run with less number of AGVs without losing a lot from the production rate, because the AGVs are being used more efficiently now. Percentage of time spent while moving with or without parts, and waiting for tracks decrease. The AGVs were never waiting for machines anyway, since the capacity of the input buffers of workcenters were always enough to accept parts immediately upon arrival.

Figure 30 suggests that the utilization of the loading/unloading workcenter increased up to almost 100%, with less idle time when compared to the beginning of the simulation when the AGV routing decisions were being made randomly. There were too many parts in the system traveling inefficiently, resulting in high flow times, and the loading/unloading workcenter was sitting idle because it was out of palettes 10% of the time.

It can be observed from Figures 31 and 32 that workcenters 2 and 4 experienced some blockage initially. This is due to AGVs not being able to pick parts up quickly enough, and the output buffers (which is of capacity 1 for WCs 2 through 4) not becoming available by the time the workcenters get done processing the next part. As the automata converge to better decisions, the blockage in both machines disappear, and this percentage is added onto the productive time.

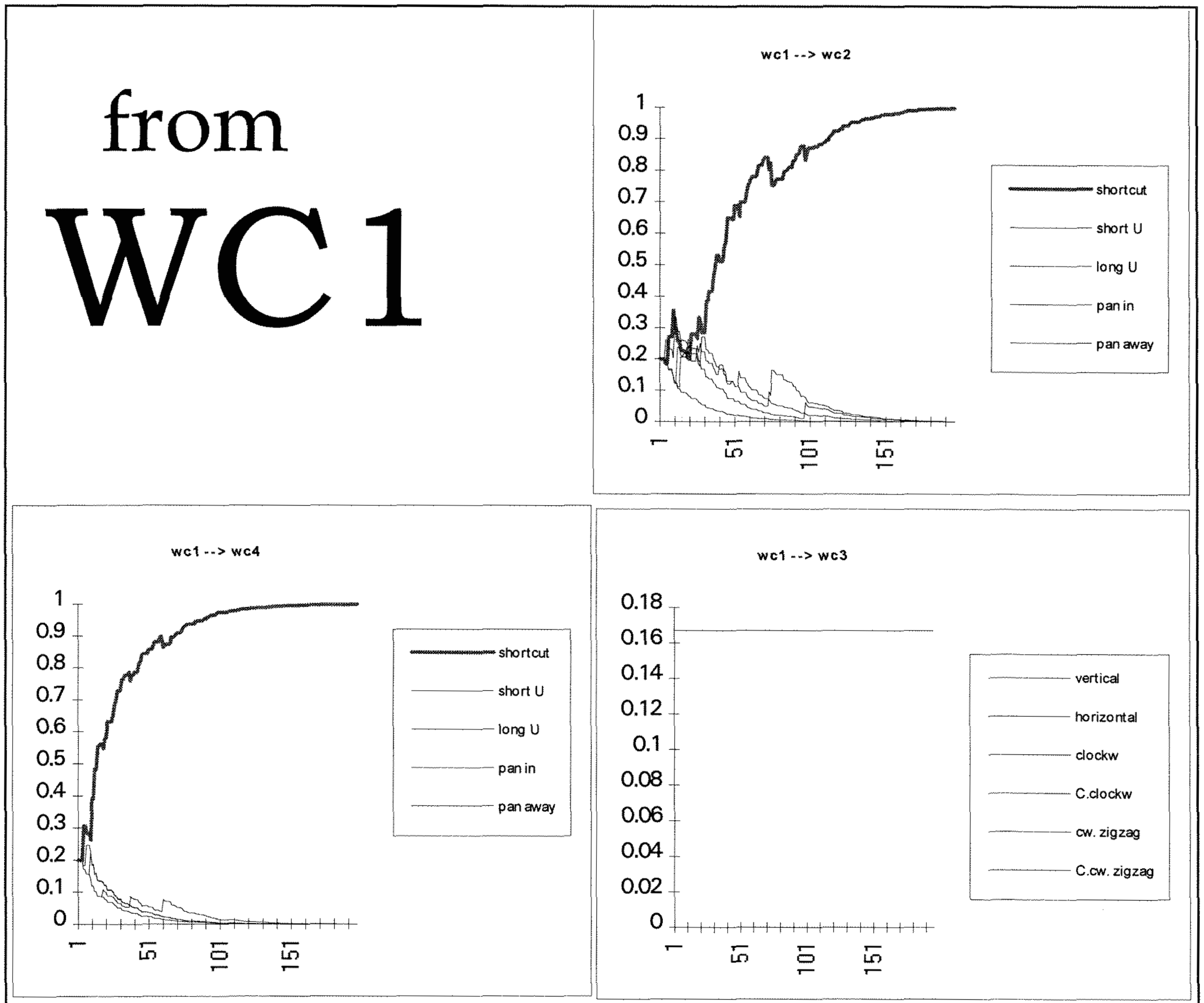


Figure 21: Automata trace charts showing convergence for routing AGVs out of WC#1.
Abscissa is the batch number, and ordinate shows the probability of making the corresponding decision.

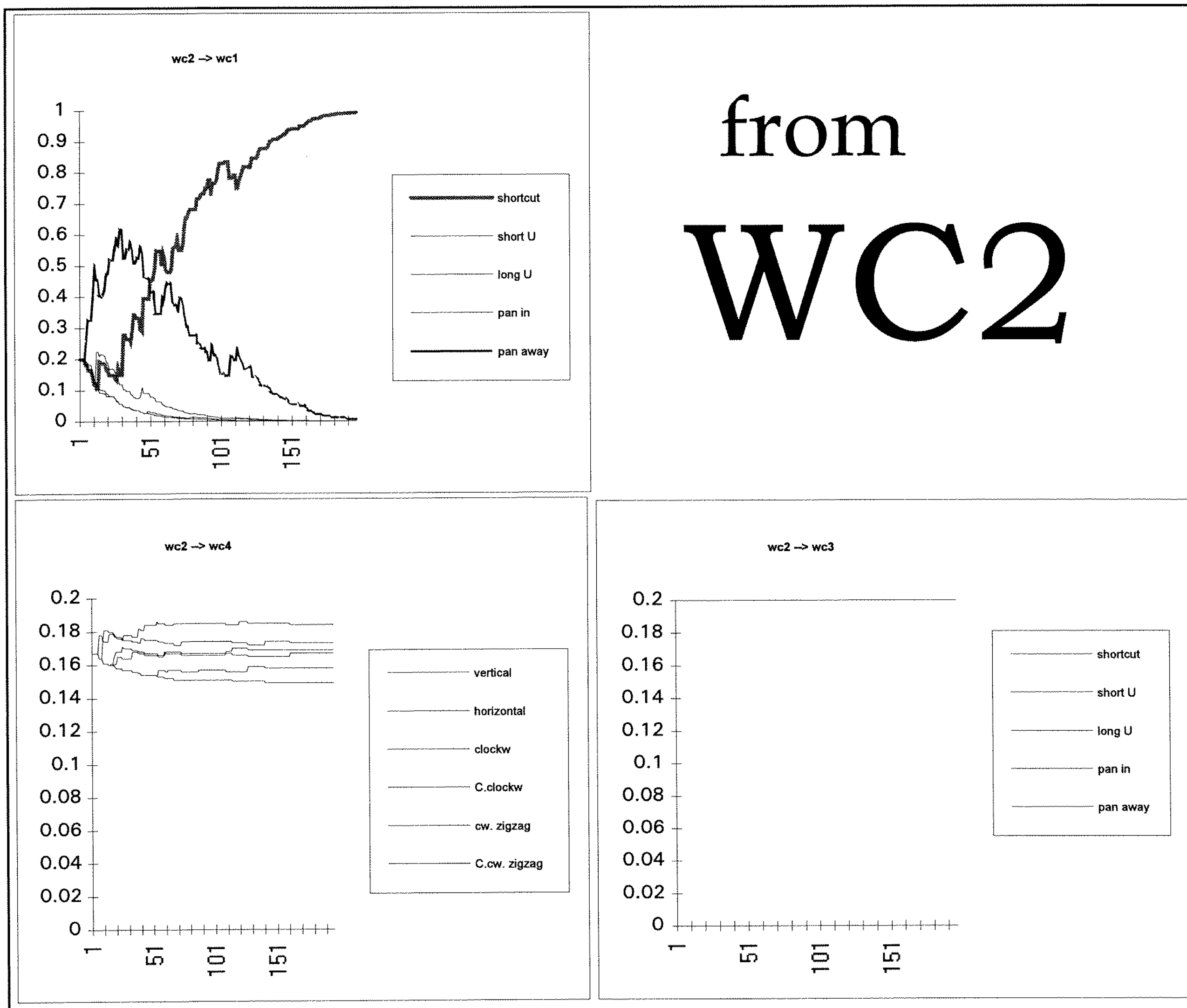


Figure 22: Automata trace charts showing convergence for routing AGVs out of WC#2. Abscissa is the batch number, and ordinate shows the probability of making the corresponding decision.

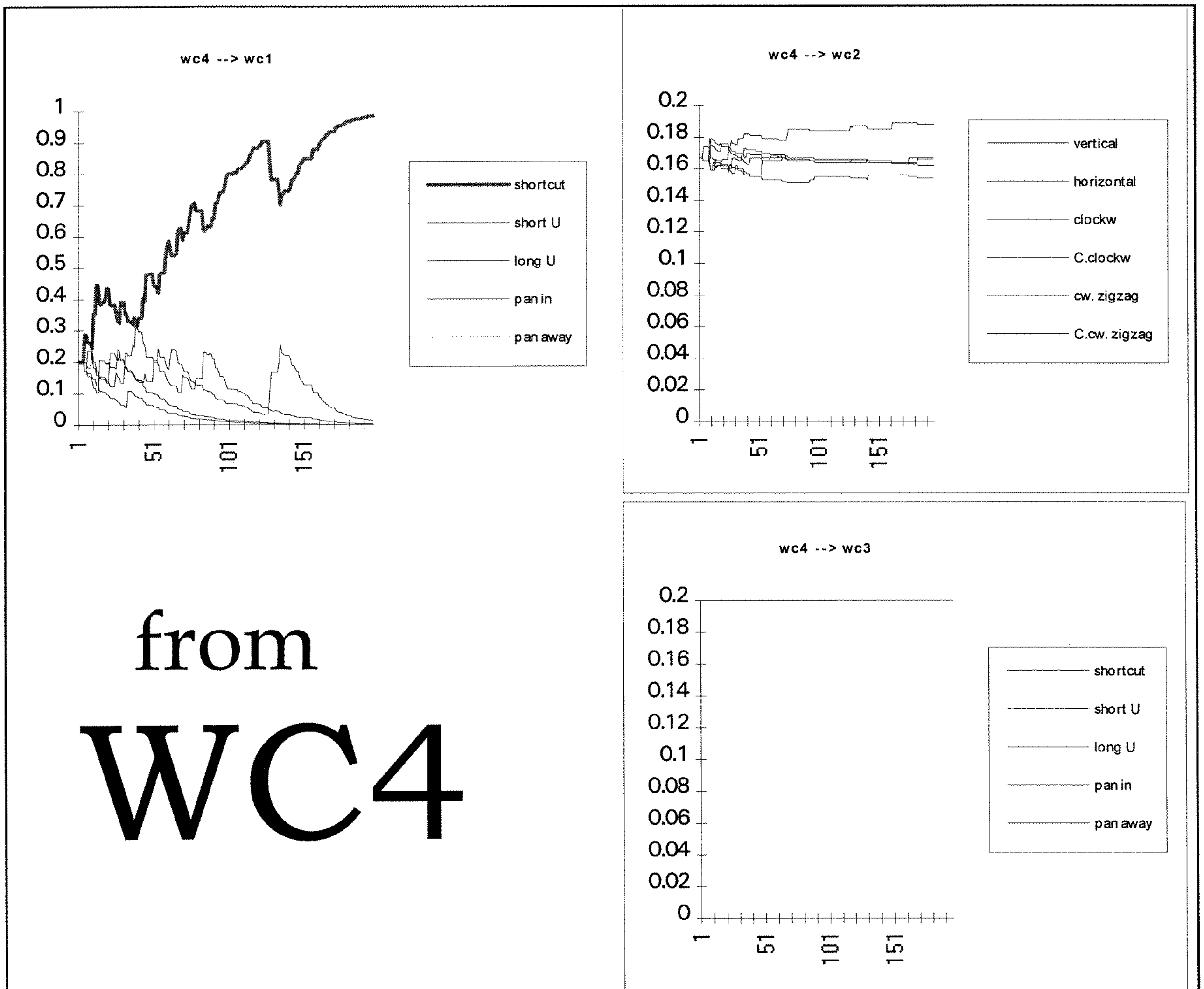
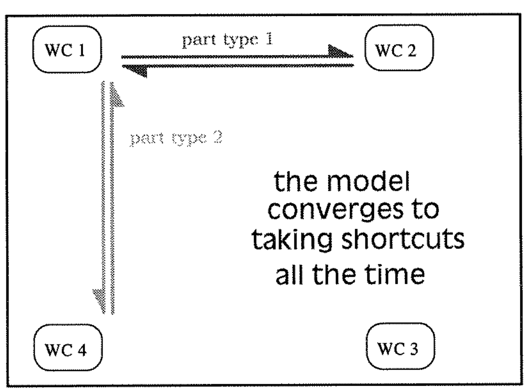


Figure 23: Automata trace charts showing convergence for routing AGVs out of WC#4. Abscissa is the batch number, and ordinate shows the probability of making the corresponding decision.

Figures 21, 22 and 23 suggest the overall convergence shown on the right.



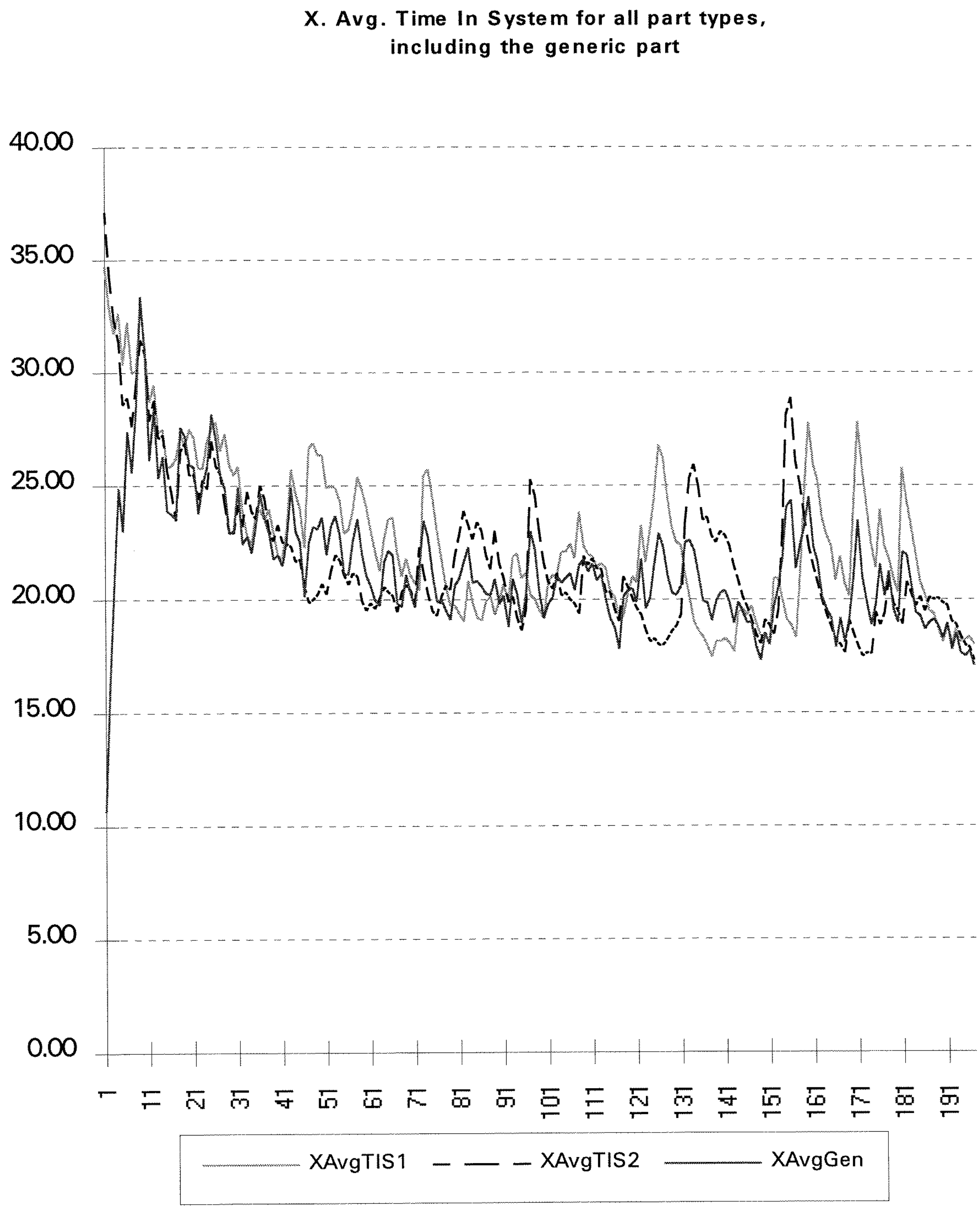


Figure 24: Exponential averages of part flow times.

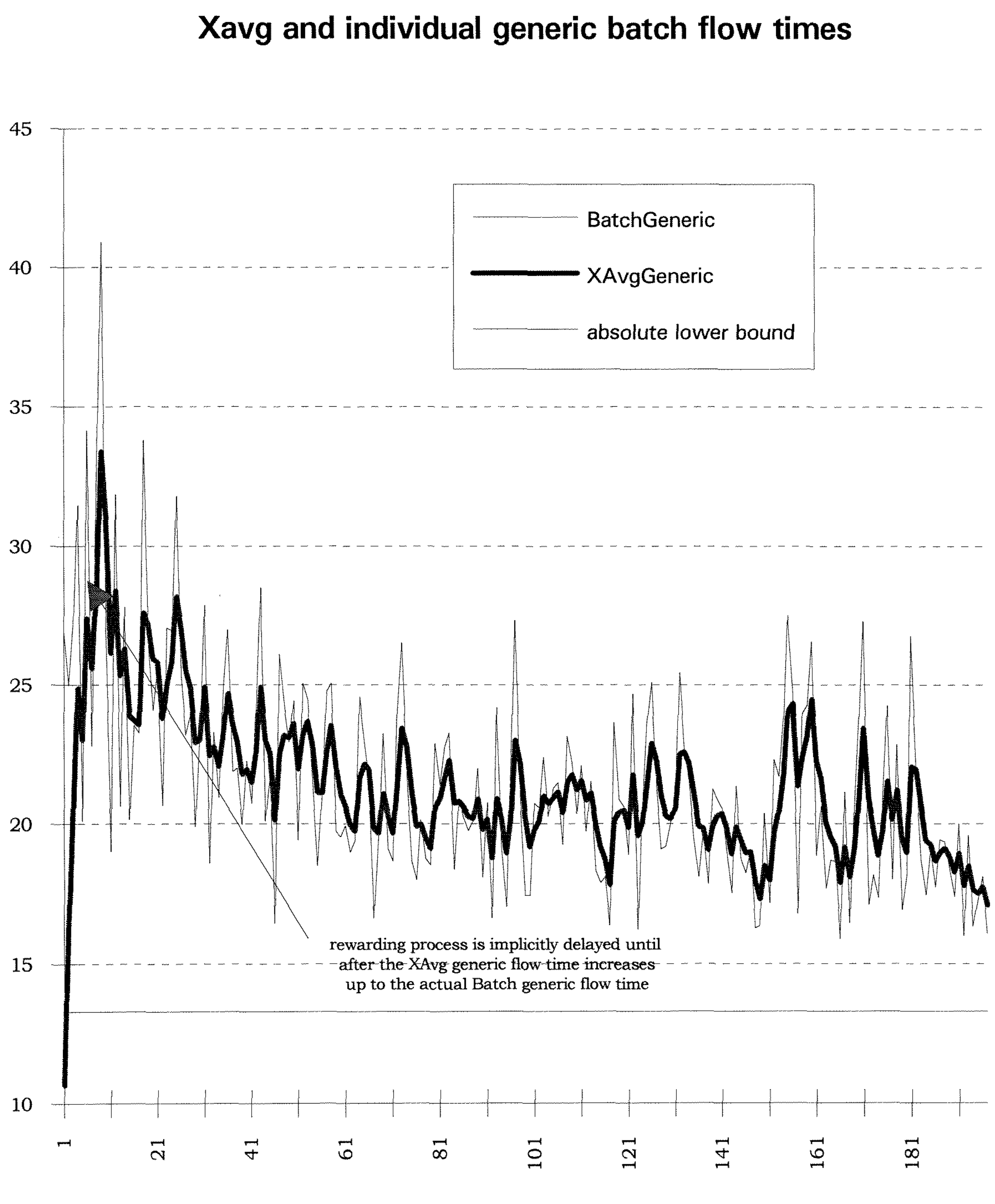


Figure 25: Individual and Exponentially smoothed generic flow times.

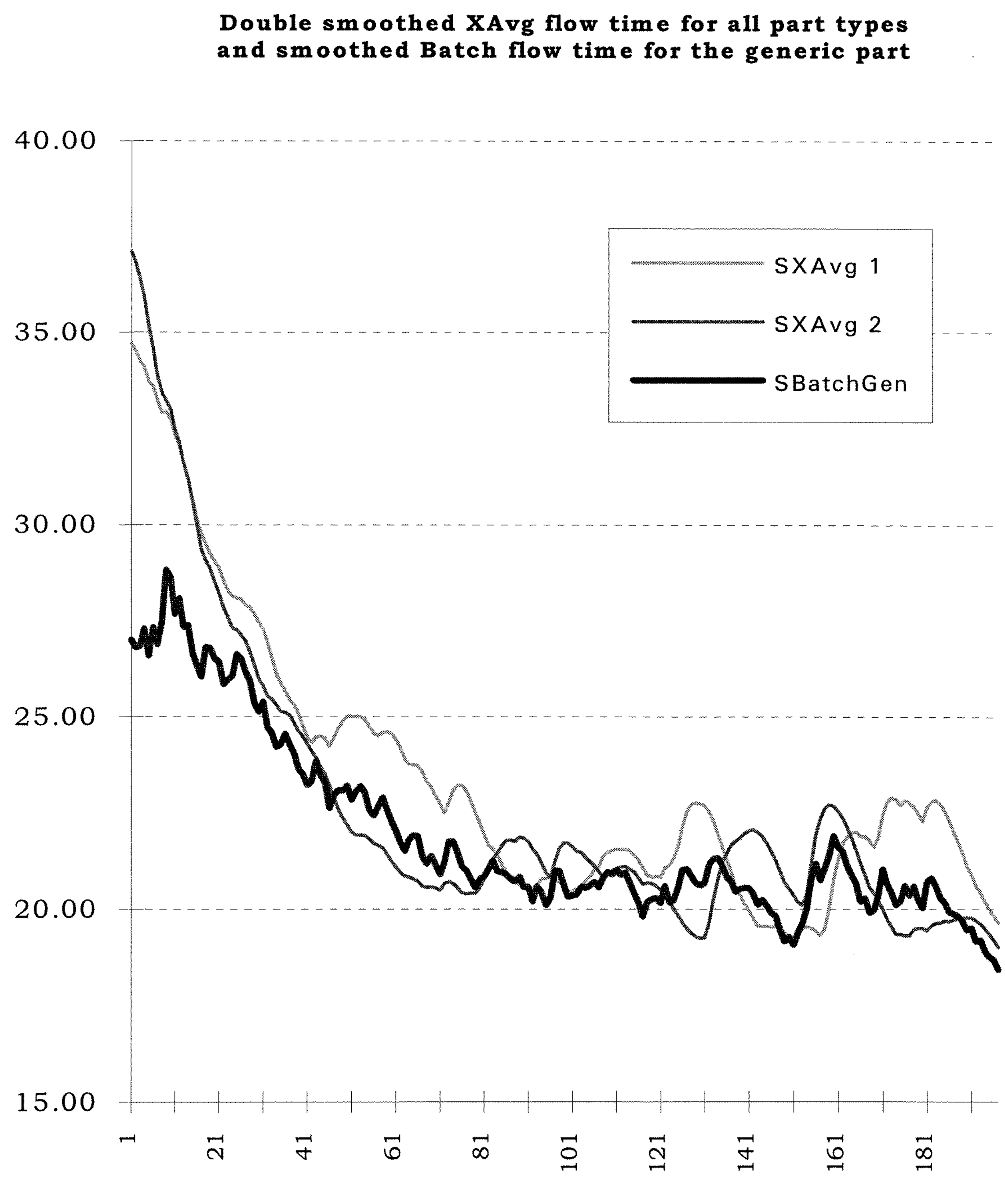


Figure 26: A smoothing constant of 0.1 is used to generate the double-smoothed flow times for all part types and the smoothed flow time of the generic part type.

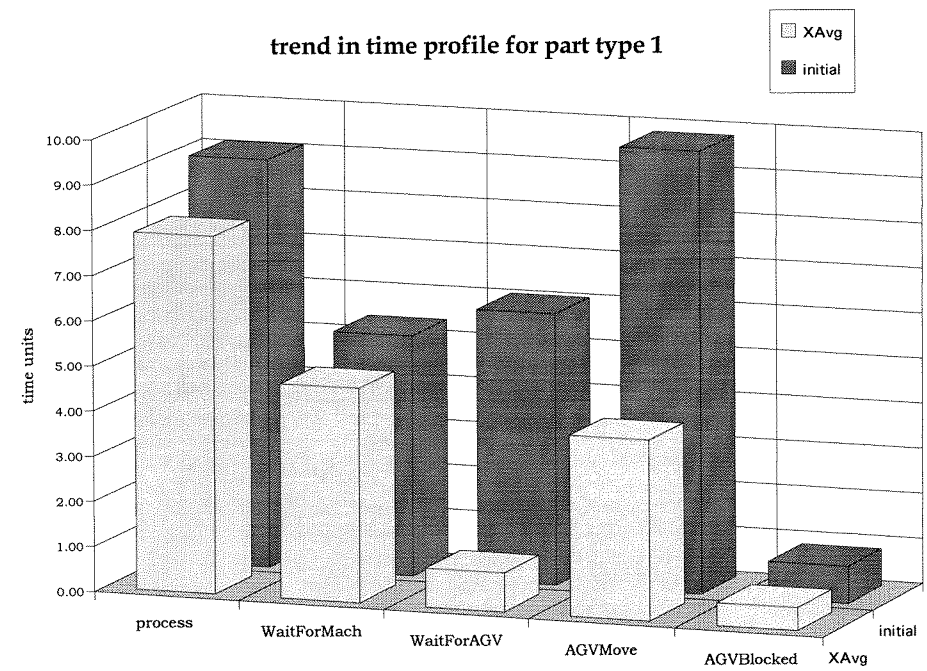


Figure 27: Approximate *absolute* time profile analysis for part type 1.

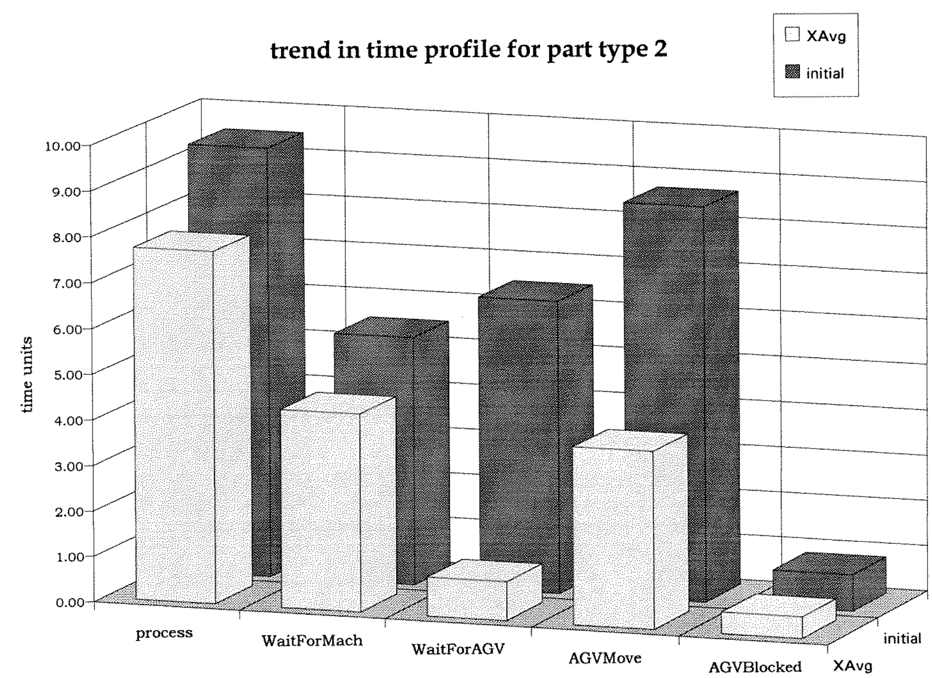


Figure 28: Approximate *absolute* time profile analysis for part type 2.

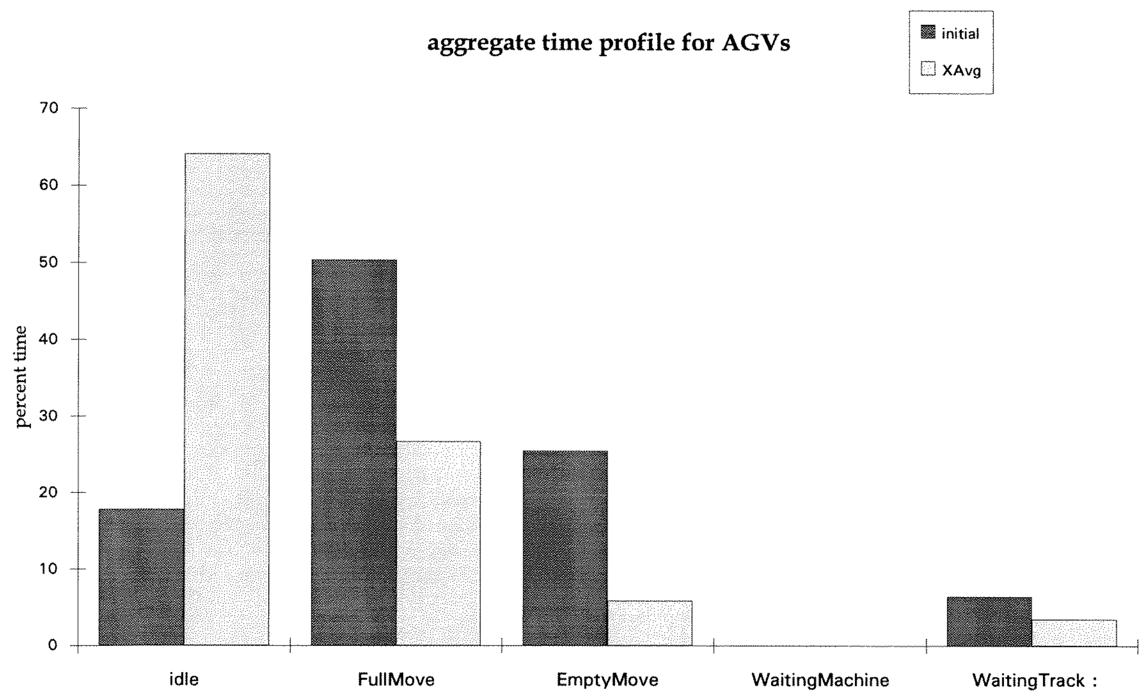


Figure 29: Trend analysis for aggregate AGV time profile.

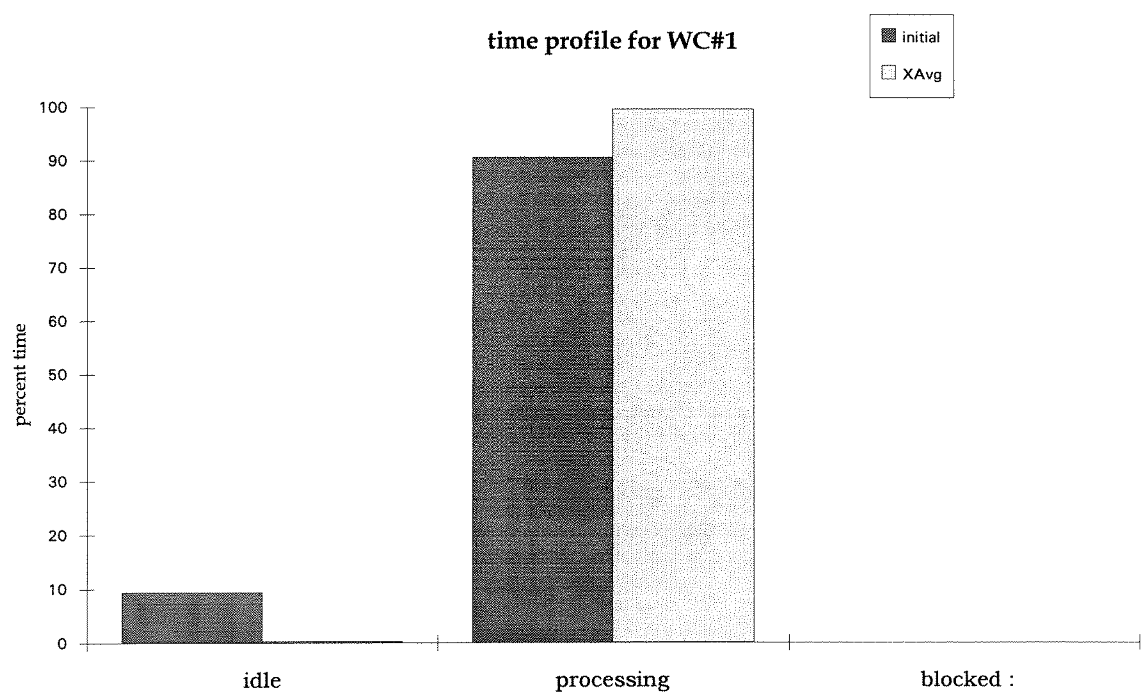


Figure 30: Trend analysis for WC#1 time profile.

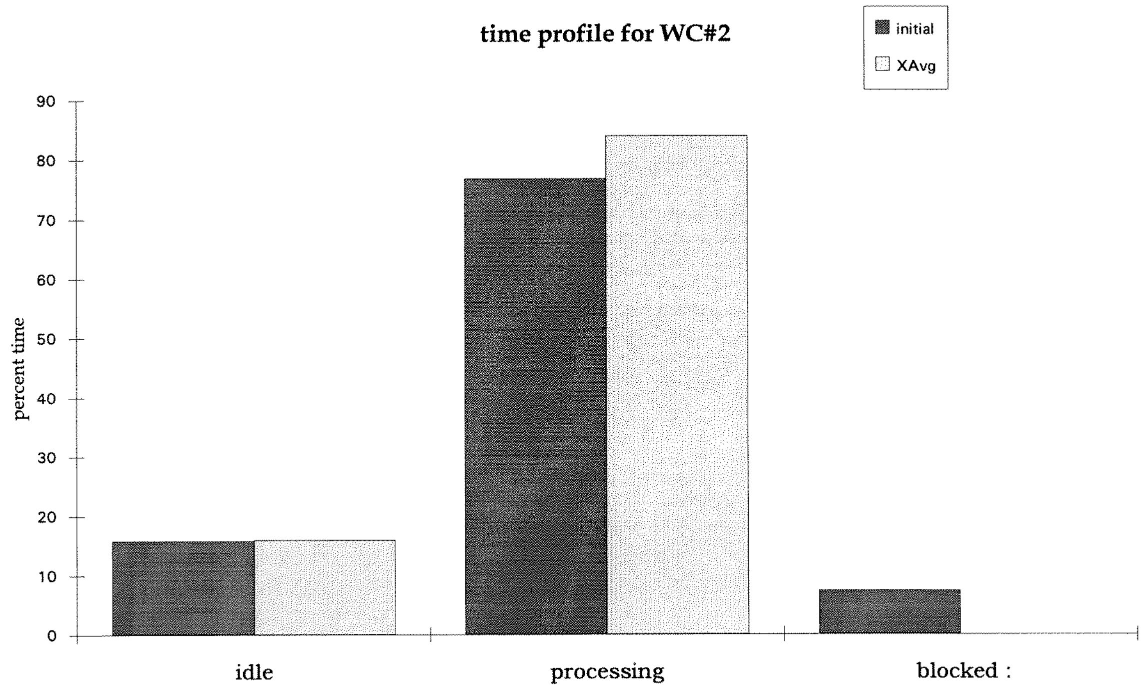


Figure 31: Trend analysis for WC#2 time profile.

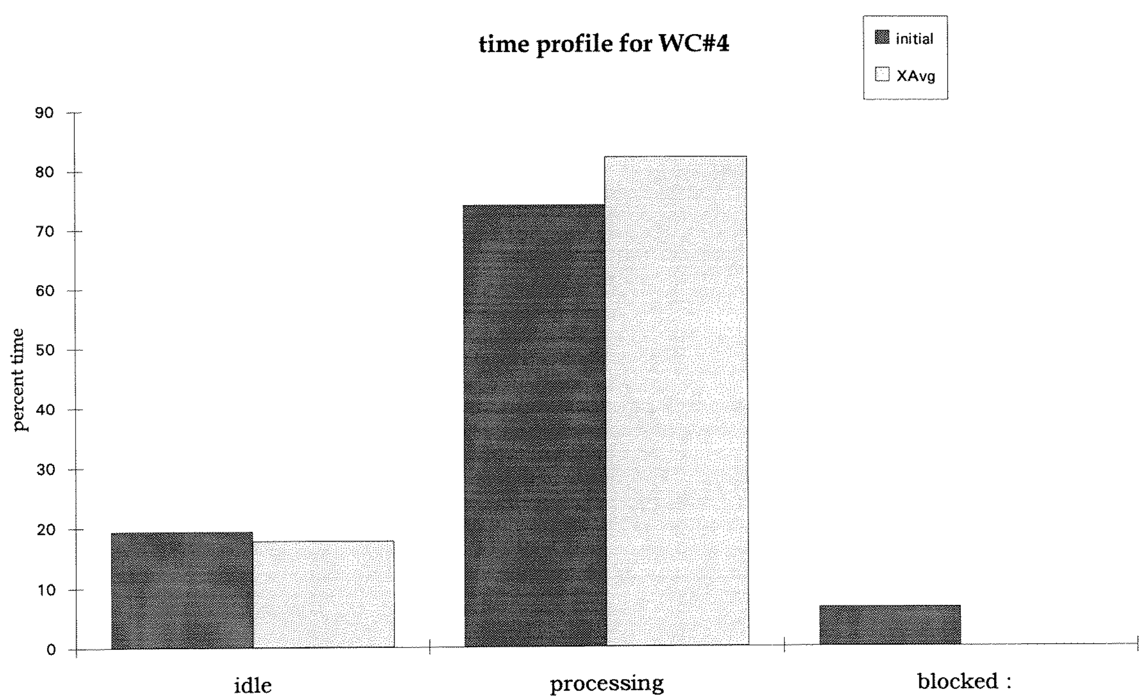


Figure 32: Trend analysis for WC#4 time profile.

V.B. Run 2: One Part Type

The inputs for the second case study are given in Exhibit 12. The FMS is run with 10 palettes and 8 AGVs, to manufacture the only part type loaded from the parts file in Exhibit 13.

this is the clever simulation model - 940410	
system inputs :	
Number of palettes =	10
Number of AGVs =	8
AGV speed =	4368.933
AGV time per track segment =	1.5
number of warmup batches =	0
XAvgTISconst =	0.005
GeneralLearningConst =	0.004
Decision Batch Size =	75
XAvgConstStats =	0.1
parts file name = try2.par	
report file (THIS FILE) name =	res\t2!4!100.out

Exhibit 12: Run 2 - input parameters.

parts file comment :: parts_trial2		
number of different types of parts = 1		
part type # 1 comment = part_type_1		
Percent arrival = 1.000000		
Process records : (WC# ; time)		
	1	1.5
	2	2.2
	1	1

Exhibit 13: Run 2 - parts file.

Because of the memory management bug in MODSIM II, this scenario was only run until the 98th batch and terminated before complete convergence of the critical automata (namely, the two automata in charge of AGV routings from WC#1 to WC#2 and from WC#2 back to WC#1). A follow-up session was run feeding in the probability vectors saved at the end of the first session, and it was confirmed that the intelligent model converges to a loop pattern, using shortcuts to go from WC#1 to WC#2 but taking the short U-turn while coming back from WC#2 to WC#1.

The automata trace charts for the first 98 batches are shown in Figures 33 and 34. They provide an insight to the fact that the convergence mentioned above would have been a natural consequence of the patterns formed, in case the model had been able to run longer.

Figure 35 shows the decreasing trend in the exponentially smoothed generic flow time, and individual batch flow time values varying randomly around their exponential average. Figure 36 is a double-smoothed graphic that demonstrates the downward trend in the exponential average of flow times for the part type and the smoothed batch generic time.

The general trend in the parts' time consumption activities (percentage distribution of flow times) is demonstrated by making use of the same method as in Section V.A. The only difference is that the absolute part time profile analysis can be performed relatively accurately, since this scenario involves the production of a single part type, and average flow time for the first ten batches can be calculated using the generic batch flow time values logged to the output file (as opposed to merely using the exponentially smoothed value at the end of the 10th batch). However, it should be kept in mind that there will *still* be some discrepancies due to the fact that the two sets of values are *not* collected in a perfectly compatible way[13].

Results of the trends in part time profile, and similar analyses for aggregate AGV and workcenter profiles for WC#s 1 and 2 are represented in Figures 37 through 40.

It can be observed from Figure 37 that there are significant decreases in all categories that the part spends time in, except for the net processing time, which should remain constant anyway.

[13] different smoothing constants are used while calculating exponential averages

Figure 38 shows the aggregate percentage time profile for AGVs. As the AGVs are used more efficiently, the need for this many AGVs decrease, and overall idle time increases while all other time categories decrease.

The utilization of the loading/unloading workcenter (WC#1) increased almost up to 100%, since the flow time for parts decreased, the number of parts wasting their time going around the system decreased, and more palettes became available for this workcenter to remain operating almost all of the time.

Although the idle time for WC#2 increased, this percentage is not deducted from the productive time: the productive time was increased as well. The percentage time that this workcenter was blocked (resulting from AGVs not responding quickly enough to pick-up requests) almost disappeared, which accounts for the percentage time increase for both other categories.

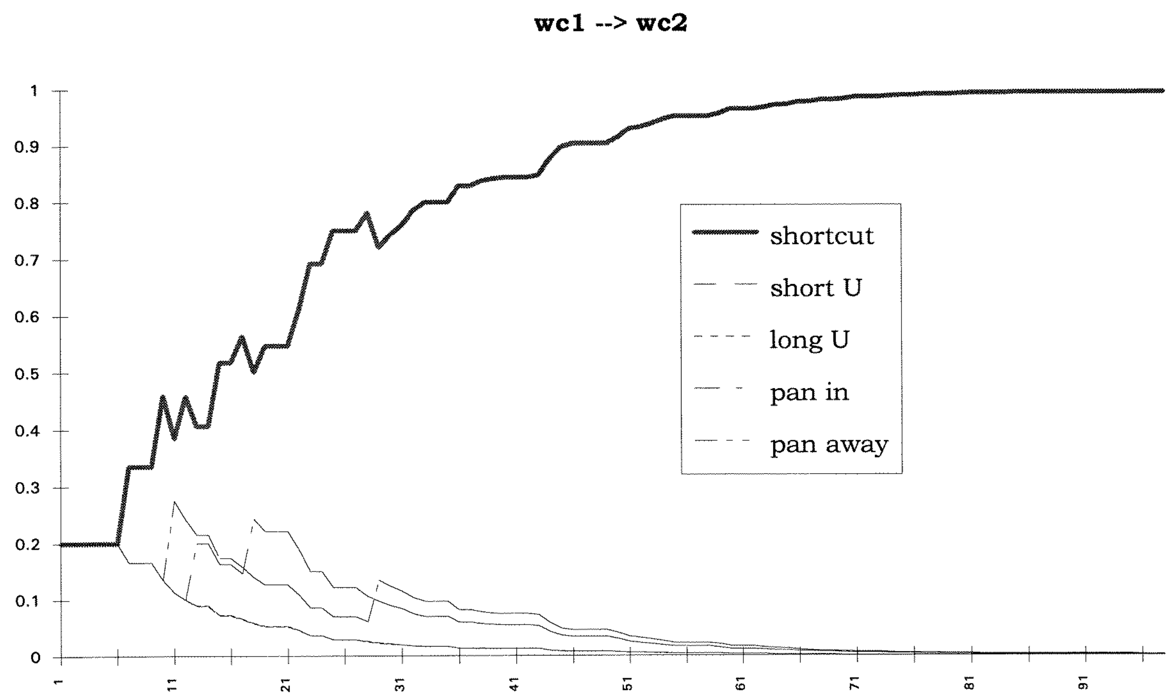


Figure 33: Automaton trace chart showing convergence for routing AGVs from WC#1 to WC#2 abscissa is the batch number, and ordinate is the probability of making the corresponding decision.

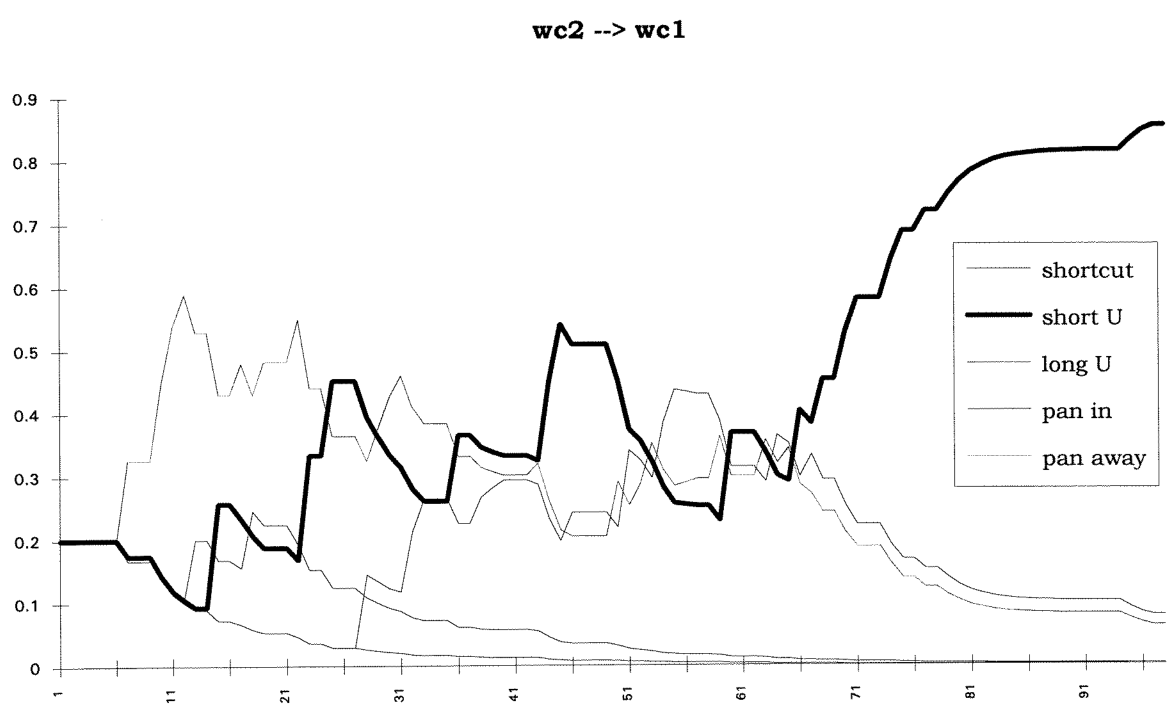
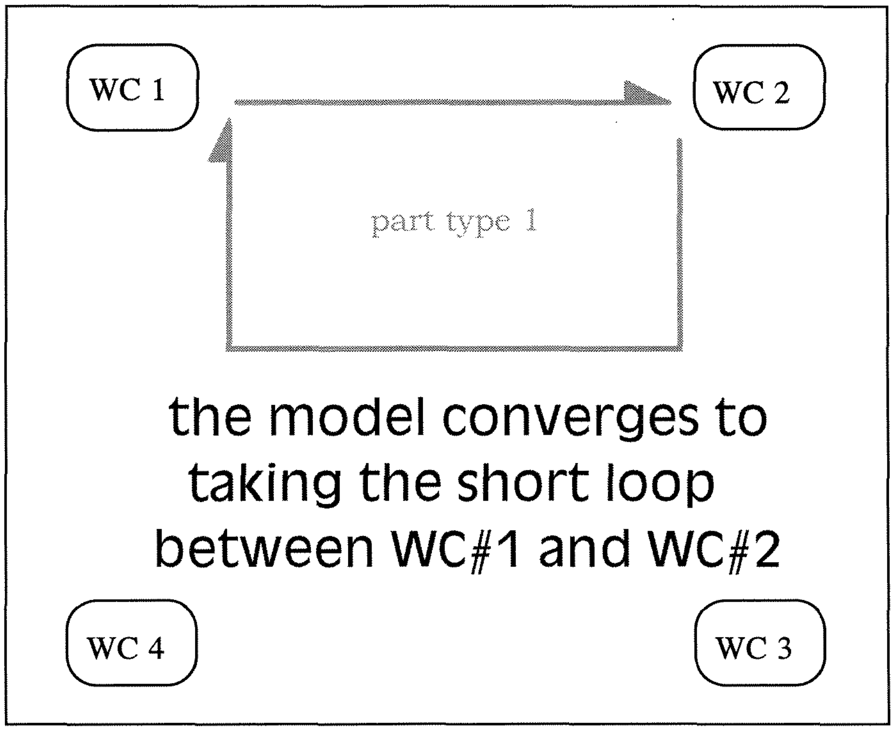


Figure 34: Automaton trace chart showing convergence for routing AGVs from WC#2 to WC#1 abscissa is the batch number and ordinate is the probability of making the corresponding decision.

Figures 33 and 34 suggest that the overall convergence is as follows:



Note that the opposite case where AGVs take the short U turn from WC#1 to WC#2 and the shortcut while coming back from WC#2 to WC#1 (a counter clockwise short loop) would have created equally beneficial overall results for the minimization of the part's average flow time.

This case study demonstrates the convergence of the intelligent model for one single replication of a simulation run. It would be possible for the automata network to converge to use the counterclockwise short loop, if different random number streams are used to come up with the uniform random numbers fed into the learning automata in the process of making dynamic random decisions.

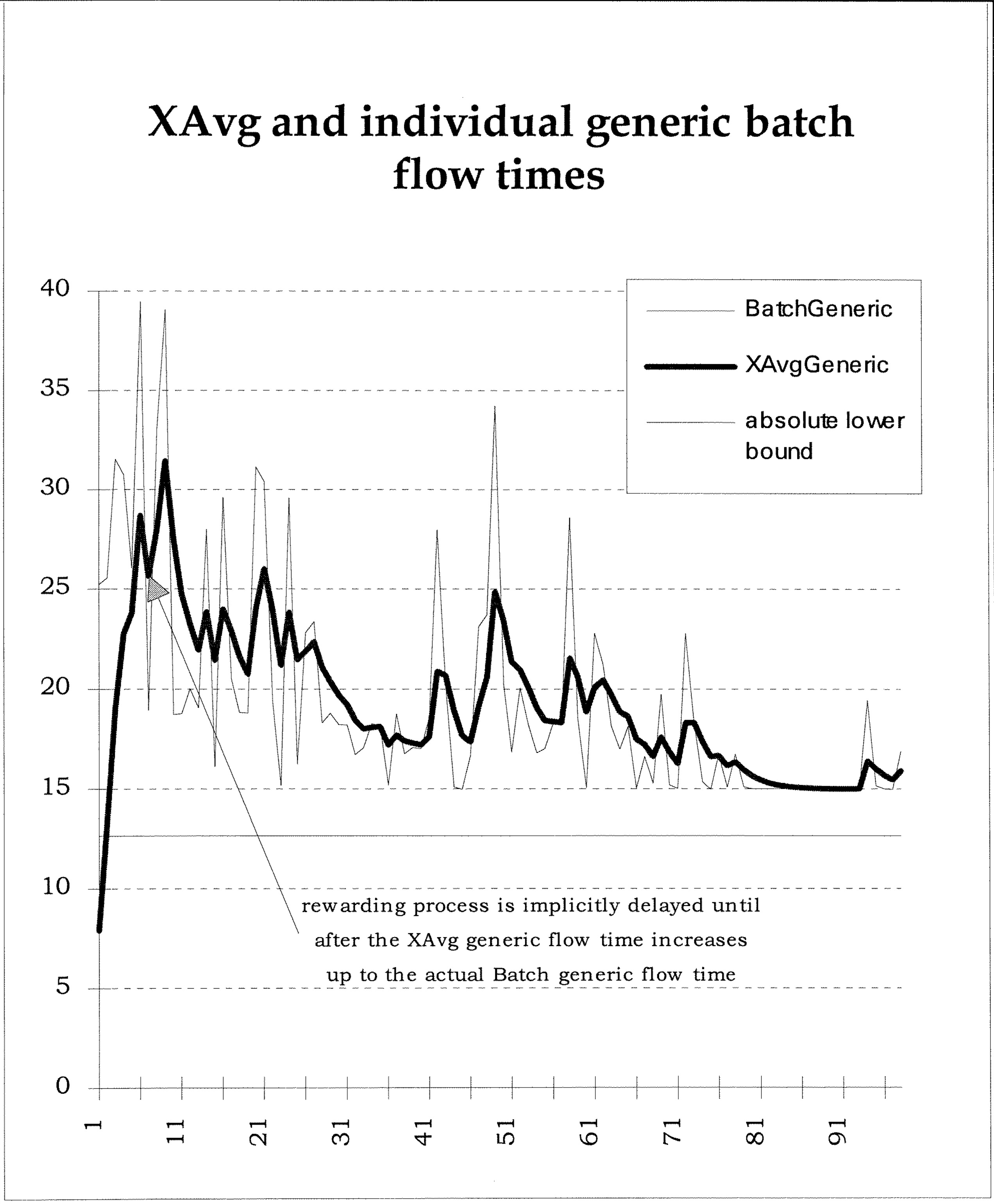


Figure 35: Batch Individual and Exponentially Smoothed generic flow time

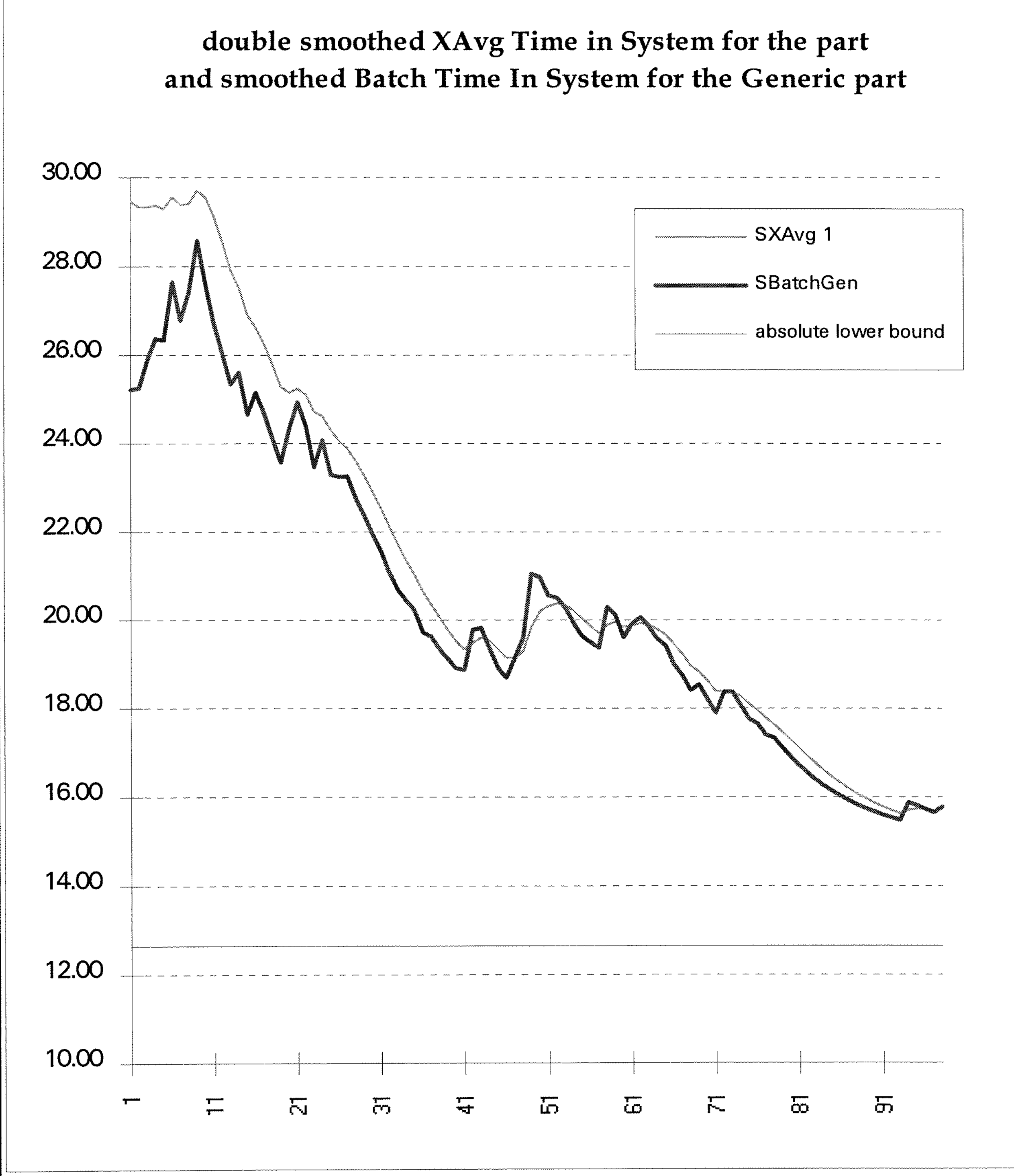


Figure 36: A smoothing constant of 0.1 is used to generate the double-smoothed flow times for the part and the smoothed flow time of the generic part type. The values are very close to each other since there is only one type of part manufactured.

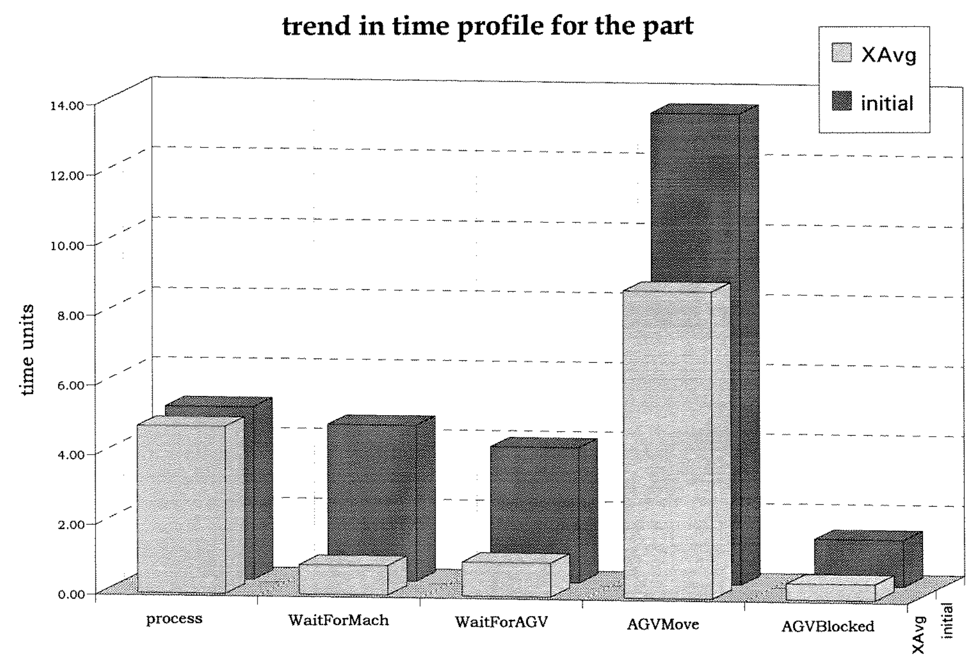


Figure 37: Approximate *absolute* time profile analysis for the part .

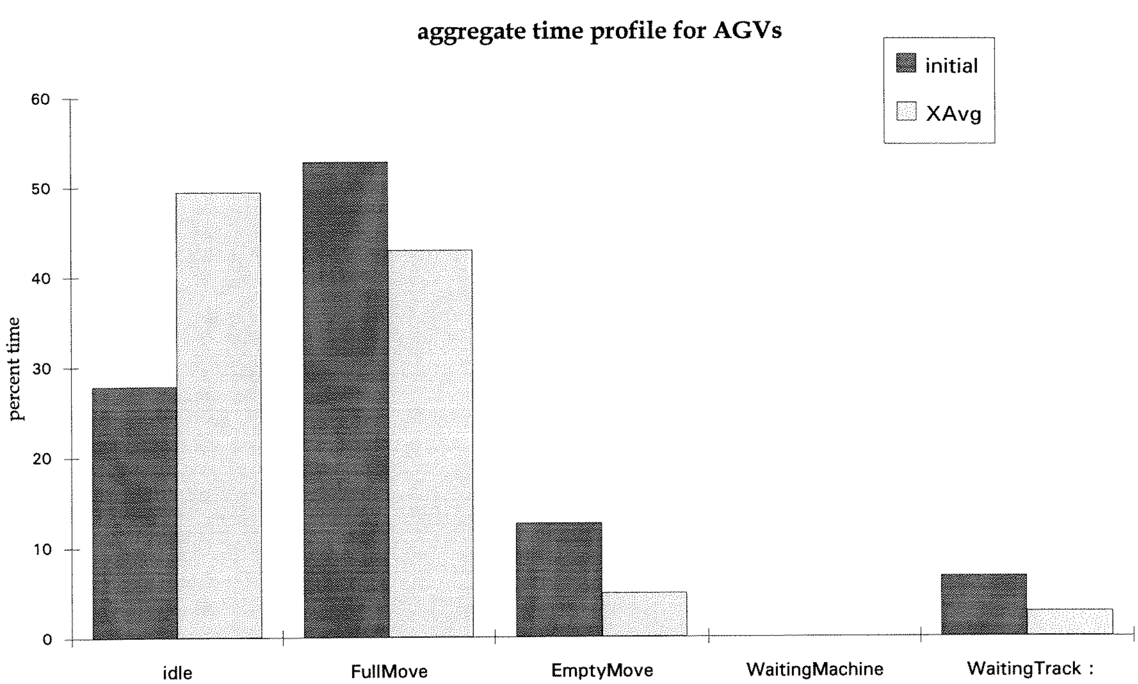


Figure 38: Trend analysis for aggregate AGV time profile.

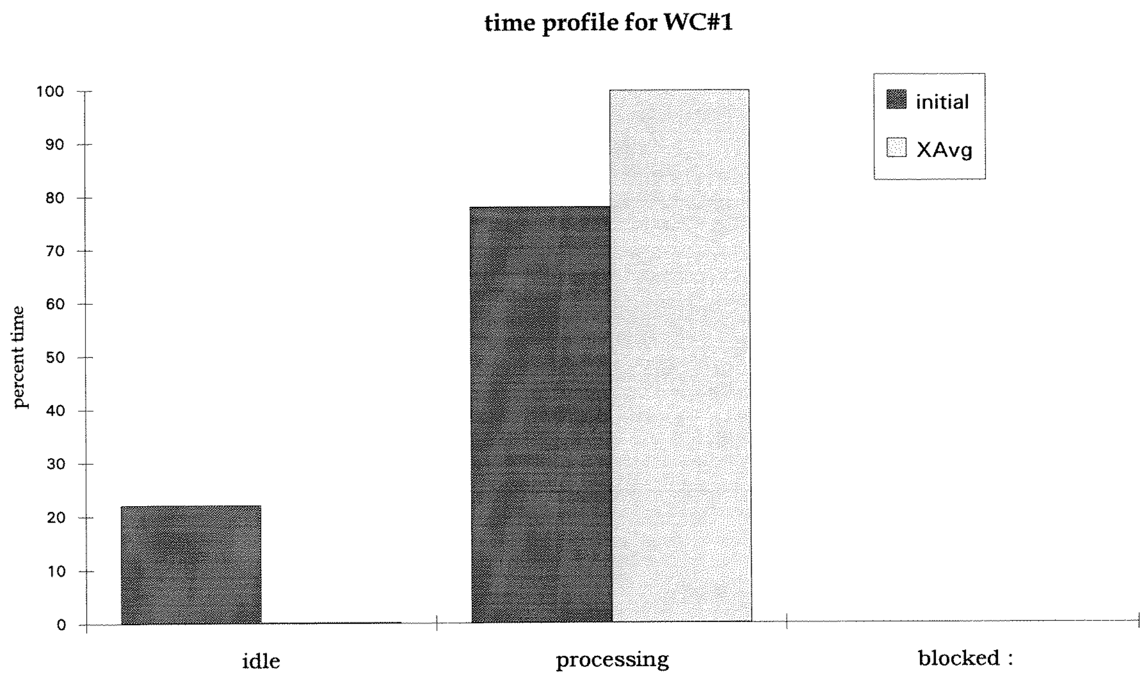


Figure 39: Trend analysis for WC#1 time profile.

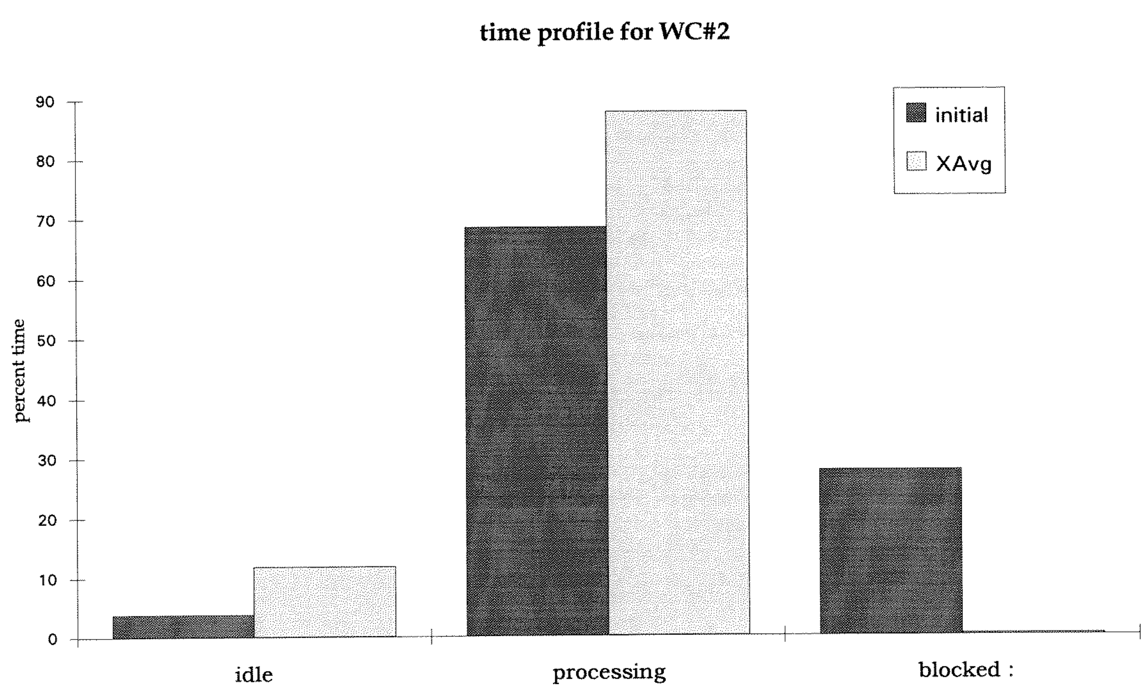


Figure 40: Trend analysis for WC#2 time profile.

VI. CONCLUSION AND FUTURE DIRECTIONS

Simulation analysis is a valuable decision making tool which is getting increasingly popular in business and academic environments as technology prospers and as decision making on tight systems gets unmanageably complex with intuition or pure analytical tools.

To benefit even more from simulation studies, analysts could make use of concepts which will introduce a prescriptive dimension to modeling. This way, simulation modeling will become a more integrated decision aid, and a more supportive experimental analysis tool.

Learning automata theory is a suitable way of introducing "intelligent", dynamic decision making units into simulation models. Synergistic benefits arise from the combination to benefit both sides, allowing analysts to build models with the capability of adaptive decision making under stochastic conditions, and providing a cost-effective, risk-free learning environment for automata.

Research in the general area of prescriptive simulation modeling and using the learning automata theory in particular should be further investigated, making use of more sophisticated software and hardware platforms.

It would also be worthwhile to integrate these kinds of simulation models with analytical tools (like mixed integer-linear programming) and heuristic rules (for instance, rules that help in job-shop scheduling, workcenter load balancing and AGV fleet sizing considerations) to gain even more overall prescriptive power, yet not sacrifice from the stochastic evaluative power of simulation modeling.

For this study in particular, as well as the general guidelines mentioned above, some immediate improvement opportunities exist along the following lines:

- turning the system into a more general, abstract FMS by converting WCs into multi-capacity locations, or even cells which contain a cluster of WCs that are capable of doing similar kinds of work
- building more detail to the AGV routing rules and more "intelligence" to the dispatcher, so as to detect full AGVs moving towards a WC to deposit parts, before allocating other AGVs that are further away from the WC that has requested a part transport

Whether it be descriptive or prescriptive; user friendliness, high runtime visual interactivity, graphical animation and dynamic graphical representation of system parameters and important statistical quantities are important points that should be kept in mind while building simulation models, in order to help make the analysts' work more enjoyable and efficient.

VII. REFERENCES

S. Benjaafar: "**Intelligent simulation for flexible manufacturing systems: an integrated approach.**" Computers & Industrial Engineering vol.22, no.3, July 1992, p.297-311.

S. V. Hoover and R. F. Perry: **Simulation, a Problem Solving Approach.** Addison-Wesley, 1990.

T. Korson, J. D. McGregor: "**Understanding Object-Oriented: A Unifying Paradigm.**" IEEE Software, Vol.33, No.9, September 1990, pp.40-60.

D.E. Monarchi, G.I. Puhr: "**A Research Typology for Object Oriented Analysis and Design.**" Communications of the ACM, September 1992, vol.35, no.9, p.35-47.

K. Narendra and M.A.L. Thathachar: **Learning Automata: an Introduction.** Prentice Hall,1989.

M. Ozden: "**Intelligent Objects in Simulation**", to appear in ORSA J. on Computing.

D.A. Popken: "**An object oriented simulation environment for airbase logistics.**" Simulation, vol.59, no.5, p.328-338.

W. D. Potter, J. A. Miller, K. J. Kochut, S. W. Wood: "**Supporting an intelligent simulation/modeling environment using the active KDL object-oriented database programming language.**" Proceedings of the Twenty-First Annual Pittsburgh Conference Part 4 (of 5). Soc. for Computer Simulation Int, San Diego, CA, 1990. p 1895-1900.

A.B. Pritsker: "**Simulation and SLAM II.**" Systems Publishing Corporation, 1984.

M. Rao, T.-S. Jiang; J.J.-P. Tsai: "**Integrated intelligent simulation environment.**" Simulation vol.54, no.6, June 1990, p.291-5

M. Rao, T.-S. Jiang; J.J.-P. Tsai, C.K. Chang: "**An intelligent simulation environment for optimal control systems.**" Proceedings of the 1987 Summer Computer Simulation Conference. SCS, San Diego, CA, 1987. p.842-4

M. Rao, Y. Ying, T.-S. Jiang; J.J.-P. Psai: "**Classifications of intelligent simulation systems.**" Proceedings of the 1990 Summer Computer Simulation Conference. SCS, San Diego, CA, 1990. p.700-2

A. H. Rutan (Ed.): **Record of Proceedings: the 22nd Annual Simulation Symposium.** IEEE, IEEE Service Center, Piscataway, NJ, 1989. 193p.

J. Snyder, G. T. Mackulack: "**Intelligent simulation environments: identification of the basics.**" 1988 Winter Simulation Conference Proceedings. SCS, San Diego, CA, 1988. p.357-63.

VIII. APPENDICES

VIII.A. Extended Bibliography for
Flexible Manufacturing Systems

VIII.B. Extended Bibliography for
Intelligent Simulation Modeling

VIII.C. Source Code Listings: Definition Modules

VIII.D. Excerpts from Implementation Code: Examples of the Reduced Semantic Gap

Appendix A:

Extended Bibliography for Flexible Manufacturing Systems

A

Title: Theoretical and experimental design of LIM in automated manufacturing systems
Author(s): Abdou, G.H.; Sherif, S.A.
Dep. Ind. Eng., Univ. Windsor, Windsor, Ont. N9B 3P4, Canada
IEEE TRANS. IND. APPLIC. VOL. 27, NO. 2 pp. 286-293 1991
Abstract: A theoretical and experimental study was conducted to assess the feasibility of employing linear induction motors (LIM'S) for automated manufacturing systems as the drive for fully flexible material handling systems MHS's. The motors were intended for use as carts to transfer parts between the three cells in a physical simulator, which consists of a load/unload cell, a manufacturing cell, and a turning cell. Experience with the currently working model confirms that the use of LIM's shows substantial promise for achieving cost and productivity gains in a wide variety of manufacturing systems.
Descriptors: factory automation; induction motors; mechanical handling; automated guided vehicles

Title: Agv System Simulation - A Planning Tool For Agv Route Layout.
Author: Andersson, M.
Corporate Source: SattControl AB, Swed
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 291-296
ISBN: 0-903608-98-7
Abstract: This paper describes a simulation package used as a tool for planning and testing of an AGV route layout. The package uses advanced computer graphics on a personal computer. This can be done in an early stage of the project.
Descriptors: *VEHICLES--*Automation; INDUSTRIAL PLANTS--Computer Simulation
Identifiers: Automated Guided Vehicle Systems (AGVS); AGV Route Layout; AGV Simulation; AGV Layout Example

Title: Proceedings of the 3rd International Conference On Automated Guided Vehicle Systems.
Author: Andersson, S. E. (Ed.)
Corporate Source: SattControl AB, Swed
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl 388p
ISBN: 0-903608-98-7

Abstract: This conference proceedings contains 35 papers discussing applications of Automated Guided Vehicle Systems (AGVS) in manufacturing processes. The conference is structured into various sessions presenting: Automated Guided Vehicle Systems (AGVS) International Overview, AGVS applications and Potential, AGVS Future trends, Systems Design, Development Trends: Sensors and Systems, AGVS in the Automative Industry, and other papers. Future trends of AGVS in the paper and printing industry, and in machining centers are described. Also discussed are AGV design for Flexible Manufacturing Systems (FMS) and the use of computer simulation as a tool for planning and testing of an AGV route layout is also presented.
Descriptors: *Vehicles--*Automation; Production Engineering--Automation; Materials Handling--Automation; Industrial Plants--Flexible Manufacturing Systems; Robots, Industrial--Vision Systems; Machine Tools--Machining Centers
Identifiers: Automated Guided Vehicle Systems (AGVS); AGVS Control Software; AGVS Safety Devices; AGV System Simulation; Eirev

Title: Colloquium On Simulation As An Aid To Plant Design.
Author: Anon
Conference Title: Colloquium on Simulation as an Aid to Plant Design.
Conference Location: London, Engl Conference Date: 1986 Mar 5
Source: IEE Colloquium (Digest) n 1986/31. Publ by IEE, London, Engl var pagings
Abstract: The proceedings contains 5 papers. The topics discussed are: an introduction to simulation-opportunities and pitfalls; simulation as an aid to plant design; matching the simulation technique to the complexity of the problem; AVG system simulation - a tool for better design; visual interactive simulation modeling in plant design and planning.
Descriptors: *Computer Simulation--*Applications; Industrial Plants; Materials Handling; Computer Aided Design
Identifiers: Visual Interactive Simulation; Automatic Guided Vehicles; Eirev

Title: Eleven Steps To Simulating An Automated System.
Author: Anon
Source: Modern Materials Handling v 39 n 16 Nov 5 1984 p 45-47
Publication Year: 1984
ISSN: 0026-8038
Abstract: Simulating a flexible manufacturing system (FMS) can trim installation costs substantially. Simulation tests all components of an FMS design - automatic guided vehicle systems, automated storage and retrieval systems, conveyors, robots, machine tools, and other materials handling and manufacturing equipment - before millions of dollars are spent installing the system. Computer simulation helps sell an FMS design to upper management, all before a single component is in place. This article shows how General Electric simulates an automated production system complete with flexible materials handling equipment. The 11-step procedure has a good track record of minimizing installation costs and maximizing productivity.
Descriptors: Materials Handling- Computer Simulation; Industrial Trucks -Applications; Warehouses-Automation; Conveyors-Control; Robotics- Applications; Machine Tools-Control
Identifiers: Flexible Manufacturing Systems; FMS; Automatic Guided Vehicles; Flexible Material Handling Systems; Automated Storage And Retrieval Systems

Title: Autonomous navigation in a manufacturing environment
Author(s): Arkin, R.C.; Murphy, R.R.
Sch. Inf. and Comp. Sci., Georgia Inst. Technol., Atlanta, GA 30332, USA
IEEE TRANS. ROBOT. AUTOM. VOL. 6, NO. 4 pp. 445-454 1990
Flexible manufacturing systems (FMS) that incorporate transport robots are currently dominated by the use of automatic guided vehicles (AGV's). These AGV's generally require significant restructuring of the workplace in order for them to be useful. The concept of flexibility in manufacturing is somewhat compromised by this strategy. This paper presents the motivation, simulation studies, and experimental results demonstrating the feasibility of migrating schema-based navigation into a FMS. In particular, the creation of a docking motor schema to accomplish interaction with the workplace is detailed.
Descriptors: flexible manufacturing systems; automated guided vehicles; navigation

Title: Interactive GPSS-PC program generator for automated material handling systems
Author(s): Ashayeri, J.; Gelders, L.F.
Katholieke Univ. Leuven, Celestijnenlaan 300A, B-3030 Leuven-Heverlee, Belgium
INT. J. ADV. MANUF. TECHNOL. VOL. 2, NO. 4 pp. 63-77 1987
Abstract: The purpose of this paper is to describe an interactive microcomputer GPSS simulation program generator for automated material handling systems. The program is written in Pascal and consists of several modules to capture data, build the model, and generate the corresponding GPSS simulation program for automated guided vehicle systems as well as surge systems. The application of the program to a real life project is then used to highlight practical advantages of this approach. The paper also presents current research (e.g. inclusion of an automated storage and retrieval system) and the degree to which such software can be extended.
Descriptors: simulation; mechanical handling; automation; computer aided manufacturing

B

Title: Integrated engineering workstation for automated guided vehicle systems design.
Author: Bakkalbasi, Omer; Gong, Dah-Chuan; Peters, Brett A.; Goetschalckx, Marc; McGinnis, Leon F.
Corporate Source: Georgia Inst of Technol, Mat Handling Research Cent, Atlanta, GA, USA
Conference Title: Proceedings of the Thirteenth Annual International Computer Software & Applications Conference - COMPSAC 89
Conference Location: Orlando, FL, USA Conference Date: 1989 Sep 20-22
Source: Proceedings - IEEE Computer Society's International Computer Software & Applications Conference. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat n 89CH2743-3), Piscataway, NJ, USA. p 783-785
ISSN: 0730-6512

Abstract: A method for performing preliminary, or rough-cut, design and analysis of automated guided vehicle (AGV) systems in which a large number of alternatives can be examined is described. The procedures are based on analytic models and are combined into an integrated engineering workstation environment. Also included in the workstation is a simulation/animation module which is used after the preliminary analysis is performed. The resulting system allows a complete study of an AGV system to be accomplished. 9 Refs.

Descriptors: *Computer Software--*Software Engineering; Materials Handling--Automation; Systems Engineering

Identifiers: Automated Guided Vehicle (AGV) Systems; Engineering Workstations

Title: Reduced-order, extended Kalman filter for AGV navigation

Author(s): Banta, L.E.; Dickerson, S.L.; Bohlander, R.; Holcombe, W.

West Virginia Univ., Morgantown, WV

American Society of Mechanical Engineers, 108th Winter Annual Meeting 8745007 Boston, MA (USA) 13-18 Dec 1987 ASME

ASME, 22 Law Drive, P.O. Box 2900, Fairfield, NJ 07007-2900 (USA), Papers and proceedings volumes available airfield, NJ 07007-2900 (USA) Paper No. 87-WA/DSC-32

Descriptors: Civil And Mechanical Engineering

Title: Decentralized control of automated guided vehicles on a simple loop.

Author: Bartholdi, John J. III; Platzman, Loren K.

Corporate Source: Georgia Inst of Technology, Atlanta, GA, USA

Source: IIE Transactions (Institute of Industrial Engineers) v 21 n 1 Mar 1989 p 76-81

Publication Year: 1989

ISSN: 0740-817X

Abstract: The heuristic is stated and analyzed from the point of view of a single automated guided vehicle (AGV) and generalized to multiple AGV's. Simulation results are consistent with the analytic results: First Encountered First Served (FEFS) works quite well for AGV's travelling a simple loop. In simulations of systems with 4-30 stations experiencing Poisson arrivals roughly matched to the aggregate capacity of the AGV's, FEFS consistently outperformed alternative simple AGV control rules. These alternative rules included first-come-first-served (mentioned in the first section), pick-up-tote-at-the-longest-queue, and pick-up-tote-closest-to-its-destination. Not only did FEFS impose less waiting time on totes (on the average), but it also reliably delivered all totes within 2% of a lower bound on the theoretical minimum time required. 4 Refs.

Descriptors: *SCHEDULING--*Mathematical Models; VEHICLES--Scheduling; CONTROL SYSTEMS, OPTIMAL; DECISION THEORY AND ANALYSIS

Identifiers: DECENTRALIZED CONTROL; AUTOMATED GUIDED VEHICLES; SIMPLE LOOP TRACK

Title: Designing The Control Of Automated Factories.
Author: Beadle, R. B.
Corporate Source: BLSL Inc, USA
Publication Year: 1986
ISBN: 0-948507-33-0
Abstract: The technique of visual interactive simulation is introduced. The advantage of this technique is demonstrated by examples that have been chosen to illustrate the particular relevance of visual interactive simulation to today's automated production units. (Author abstract)
Descriptors: *Computer Simulation--*Industrial Applications; Industrial Plants--Automation; Conveyors--Computer Applications; Welding--Automation; Computer Graphics--Interactive; Automobile Plants--Automation
Identifiers: Automated Factory Control; Visual Interactive Simulation; Computer-Controlled Conveyor Network; Automatic Welding Facility; Automatic Guided Vehicle (AGV)

Title: Frito-Lay Takes A Fresh Approach To Automation.
Author: Beck, Larry
Source: Modern Materials Handling v 42 n 1 Jan 1987 p 78-81
ISSN: 0026-8038
Abstract: Frito-Lay, Inc. gets new products onto store shelves quickly and fills orders with high accuracy. Supporting this quick response capability and outstanding service to sales are 40 manufacturing facilities nationwide. The newest is in Kern County, Calif. , near Bakersfield, and includes the company's most automated warehousing and distribution operation. The new system handles cases of potato chips and corn products from an adjoining production area. The system was designed with the following criteria in mind: automated control and tracking of inventory to provide timely shipments and fresh products; gentle, consistent handling to reduce breakage; and flexibility and expandability to ensure continued good service to sales and quick response to the introduction of new products.
Descriptors: *Materials Handling--*Automation; Food Products--Inventory Control; Computer Simulation; Warehouses--Automation
Identifiers: Frito-Lay Inc.; Automatic Guided Vehicles; Case Handling; Automated Storage And Retrieval Systems

Title: Why a single aisle miniload system is not simple to model
Author(s): Bengtson, N.M.; Gomez, R.J,; Abrams, M.A.; Haigh, P.L.; Comfort, J.C. (eds.)
Dep. Comp. Sci., North Carolina State Univ., Raleigh, NC 27607, USA
1988 Winter Simulation Conference. Pp. 603-608 1989
1988 Winter Simulation Conference 12-14 Dec 1988 San Diego, CA (USA)
Publ: Society For Computer Simulation International, P.O. Box 17900, San Diego, Ca 92117 (Usa),
Abstract: In order to test the performance of various storage assignment policies of a miniload warehouse system a general simulation model was developed. Even though the system concept was relatively simple, the simulation model became quite complex. This is because there are numerous factors in the design and operation of even a simple, single aisle system which influence performance. This inherent complexity has contributed to the difficulty in anticipating the performance of large, automated warehouses. A set of assumptions is given which can realistically be expected to simplify a simulation model.
Descriptors: storage; warehousing; simulation; plant and factory layout; modeling; automated guided vehicles; distribution

Title: Free ranging AGV systems: Promise, problems and pathways
Author(s): Besant, C.B.; Broadbent, A.J.; Premi, S.K.; Walker, S.P. Imperial Coll., UK
IFS Advanced Manufacturing Summit '85; AUTOMAN '85 8520476 Birmingham (UK) 14-17 May 1985
Descriptors: Civil And Mechanical Engineering; Electrical Engineering; General Engineering And Technology; Mathematics Section Heading: Civil And Mechanical Engineering ; Electronics Engineering; General Engineering And Technology ; Mathematics

Title: Modeling An AGV Automobile Body-Framing System.
Author: Bookbinder, James H.; Kotwa, Terrence R.
Corporate Source: Univ of Waterloo, Waterloo, Ont, Can
Source: Interfaces (Providence, Rhode Island) v 17 n 6 Nov-Dec 1987 p 41-50
Publication Year: 1987
ISSN: 0092-2102
Abstract: General Motors of Canada is undertaking a multimillion-dollar modernization of its Oshawa, Ontario assembly plant. Automatic guided vehicles (AGVs) and a substantial amount of robotics and other automation will be used in the body-framing area. We developed a computer simulation model to estimate the minimum number of AGVs required to meet reliably the minimum targetted output. We found the framing system could produce 20 percent more cars per shift, but this increase beyond the target would require acquisition of 70 percent more AGVs. We also found that the system output was relatively insensitive to the cycle time of the automation cells and to the failure rates of three critical process stations. (Author abstract) 3 refs.
Descriptors: *Vehicles--*Industrial Applications; Automobile Plants-- Automation; Robots, Industrial; Computer Simulation; Optimization
Identifiers: Automatic Guided Vehicles (AGVs); Plant Modernization; Automation Cells

Title: Tandem AGV systems: A partitioning algorithm and performance comparison with conventional AGV systems.
Author: Bozer, Yavuz A.; Srinivasan, Mandyam M.
Corporate Source: Univ of Michigan, Ann Arbor, MI, USA
Source: European Journal of Operational Research v 63 n 2 Dec 10 1992 p 173-191
Publication Year: 1992
ISSN: 0377-2217
Abstract: In an earlier paper, Bozer and Srinivasan introduced the tandem concept for automated guided vehicle (AGV) systems and presented an analytical model to evaluate the throughput performance of a basic component of the system; namely, a single vehicle serving a set of workstations under the First-Encountered-First-Served rule. In this study, using the above analytical model and certain column generation techniques, we present a heuristic partitioning scheme to configure tandem AGV systems. The partitioning scheme is based on a variation of the well-known set partitioning problem. It is aimed at evenly distributing the workload among all the AGVs in the system. We demonstrate the procedure with two numerical examples. Using simulation, the performance of the tandem configuration obtained for each example is compared to that of the corresponding conventional AGV system. (Author abstract) 5 Refs.
Descriptors: *Vehicles; Mathematical Models; Performance; Algorithms; Computer Workstations; Set Theory; Heuristic Methods
Identifiers: Automated Guided Vehicle Systems; Set Partitioning

Title: Automatic programming of AGVS simulation models
Author(s): Brazier, M.K.; Shannon, R.E.; Thesen, A.; Grant, H.; Kelton, W.D. (eds.)
Ind. Eng. Dep., Texas A+M Univ., College Station, TX 77843, USA
1987 WINTER SIMULATION CONFERENCE. pp. 703-708 1987
1987 Winter Simulation Conference 14-16 Dec 1987 Atlanta, GA (USA)
Publ: SOCIETY for Computer Simulation, P.O. BOX 17900, SAN DIEGO, CA 92117-7900 (USA),
Abstract: This paper presents a knowledge based modeling system that allows a manufacturing engineer who has very limited knowledge of simulation methodology to quickly and correctly, develop and run a simulation model of an automated guided vehicle system (AGVS). The modeling system is capable of guiding and assisting the engineer with a level of "expertise" comparable to a trained simulation specialist. The modeling support program is an automatic programming system written in Turbo-Prolog which generates the computer code for the required model and experiment in the SIMAN simulation language.
Descriptors: production; guidance and control systems; mechanical handling; transportation; manufacturing; modeling; simulation

Title: ANTICIPATION AND SIMULATION IN WORKSHOP TRANSPORT.
Author: Buda, J.; Badida, M.; Vrlik, J.
Corporate Source: Technical Univ Kosice, Czech
Conference Title: Proceedings of the 4th International Conference on Automated Guided Vehicle Systems: AGVS-4.
Conference Location: Chicago, IL, USA Conference Date: 1986 Jun 24-26
Source: Publ by IFS (Publ), Kempston, Engl p 103-113
ISBN: 0-938507-13-6
Abstract: Systemology is becoming a prevailing approach in transport operations on the workshop level. From theoretical and practical viewpoints it is desirable to develop combinations of transport situations and needs. The paper presents approaches to elaboration of model solutions at the authors' workplace. It emphasizes the necessity to classify the transport functions and their carriers, and presents mainly the principle of anticipation development for the function and space relations. (Author abstract) 5 refs.
Descriptors: *Materials Handling--*Automation; Materials--Transportation; Vehicles--Industrial Applications
Identifiers: Workshop Transport; Transport Functions; Automated Guided Vehicle Systems (AGVS); AGVS Design

Title: Planning and implementing a flexible assembly system supported by simulation
Author(s): Bullinger, H.-J.; Sauer, H.
Fraunhofer Inst. Ind. Eng. (IAO), Silberburgstr. 119 A, D-7000 Stuttgart 1, FRG
INT. J. PROD. RES. VOL. 25, NO. 11 pp. 1625-1634 1987
International Production Management Conference 16-17 Mar 1987 Enschede (Netherlands)

Abstract: The modular assembly system for heavy fork lifts is distinguished by its flexibility and by the successful integration of all subsystems. Assembly group technology as well as an integrated material supply concept are characteristic of this solution. The planning process has been supported by the application of simulation model which allows evaluation of different solutions and layouts. The simulation of different ways of operating the system, i.e. mixed or separated assembly of particular products within the same subsystem are very helpful for judgement and decision for a particular system and its operation.
Descriptors: assembling; flexible manufacturing systems; production planning; simulation; automated guided vehicles; modular design

Title: SO YOU WANT TO BUY AN AGV SYSTEM.
Author: Burton, J.
Corporate Source: Ingersoll Engineers Inc, UK
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 57-62
ISBN: 0-903608-98-7
Abstract: A decision to invest in an AGV system has to take into account many factors regarding operation, product mix, current and future business objectives. AGV systems are, by and large, based on existing hardware and systems technology. However, they also involve a complete range of other interfaces and activities from parts scheduling and computer interfacing to special floor specifications, all of which require time, resourcing and costing. How the end user organises and plans the project has a direct bearing upon the ease with which these systems are implemented. This paper will demonstrate how careful planning, clear specifications and meticulous project organisation are crucial to the successful implementation of AGV systems. (Author abstract)
Descriptors: Vehicles-Automation; Cost Accounting; Marketing; Production Engineering-Automation
Identifiers: Automated Guided Vehicle Systems (AGVS); AGVS Purchase Planning; AGVS Implementation; Establishing Budgets; Computerised Model Simulation

C

Title: Simulation study of an automated guided-vehicle system in a Yugoslav Hospital.
Author: Ceric, Vlatko
Corporate Source: Univ of Zagreb, Zagreb, Yugosl
Source: Journal of the Operational Research Society v 41 n 4 Apr 1990 p 299-310
Publication Year: 1990
ISSN: 0160-5682
Abstract: A simulation study of an automated guided-vehicle system connected with an automated vertical transportation system in a large hospital is described. The aim of the study was to determine the allocation and size of storage space for vehicles, the minimum number of vehicles required, the accomplishment of a transportation-network timetable and the queueing situation in the system. This study is part of a simulation study of internal transportation in the new, 1100-bed University Hospital in Zagreb, Yugoslavia, which is under construction. (Author abstract) 5 Refs.
Descriptors: *Materials Handling--*Computer Simulation; Hospitals-- Transportation; Vehicles--Automation; Transportation--Scheduling
Identifiers: Automated Guided Vehicles; Developing Countries; Vertical Transportation System

Title: Modeling and optimization of a flexible manufacturing system.
Author: Chavali, R. N.; Keswani, S.; Bose, S. C.
Corporate Source: Indian Inst of Technology, Madras, India
Source: Journal of Applied Manufacturing Systems v 3 n 2 Winter 1990 p 25-27
Publication Year: 1990
ISSN: 0899-0956
Abstract: Utah State University, in Logan, Utah, is one of seven universities in the nation which offers an accredited program in manufacturing engineering. The university's Mechanical and Manufacturing Engineering Department has undertaken a research project to implement a Flexible Manufacturing System (FMS). The proposal for the development of FMS was funded by the Society of Manufacturing Engineering Education, Marriner S. Ecclles foundation at Utah, and a university faculty research grant. Utah State University is also funded by IBM to develop a master's program in Computer-Integrated Manufacturing (CIM) through the CIM for Higher Education program. The project described will have a direct impact on industries moving toward integrated manufacturing. The research findings aid in attaining a uniform machine utilization and flow of parts through the FMS cell, thereby minimizing inventory and production costs. (Author abstract)
Descriptors: *Flexible Manufacturing Systems--*Optimization; Computer Integrated Manufacturing; Flexible Manufacturing Systems--Computer Simulation; Materials Handling--Automation; Computer Workstations
Identifiers: Fms; Automated Guided Vehicles; Cim; Machine Utilization

Title: A simulation study of automated guided vehicle dispatching
Author(s): Cheng, T.C.E.
Dep. Actuarial and Manage. Sci., Univ. Manitoba, Winnipeg, Man. R3T 2N2, Canada
Robotics Comp. Integrated Manuf. Vol. 3, No. 3 Pp. 335-338 1987
Abstract: This paper reports a simulation study to investigate the relative performance of five automated guided vehicle (AGV) dispatching rules. The study found that distance-based rules are most effective while AGV Attribute-based rules are least effective. Both AGV speed and the number of AGVs in use are factors critically affecting the performance of a flexible manufacturing system. This paper demonstrates that computer simulation is a viable tool for assisting FMS design and evaluation.
Descriptors: simulation; automated guided vehicles; routing; flexible manufacturing systems

Title: Simulated intelligent flexible cell for the assembly of multi-component systems
Author(s): Cohen, G.
Journal: Engineering Applications of Artificial Intelligence vol.4, no.2 p.121-30
Publication Date: 1991 Country of Publication: UK
ISSN: 0952-1976
Abstract: An intelligent flexible cell for assembly is an autonomous, self-contained, highly productive assembly system with the ability to reason about actions and their results. Multi-component systems are represented in this paper by four different pneumatic motors, varying in power capability, size, weight and pattern, and in the method of assembly. The knowledge base of the expert system described contains facts and rules to define the flexible assembly cell and the structures of the pneumatic motors. The inference engine integrates the parameters which define the current situation of the assembly cell, with the rules and the facts in the knowledge base, to determine the next operations to be implemented. The numerical runs performed on a simulated cell address aspects associated with the design and methods of operation of flexible assembly cells: the use of gripper adaptor adjustments, the cost savings resulting from a vision system and various methods of cell operation relating to batching and loading of parts, and release and dispatching operations. (14 Refs)
Descriptors: assembling; computer vision; expert systems; flexible manufacturing systems; industrial robots; inference mechanisms; pneumatic systems
Identifiers: industrial robots; multi-component systems; intelligent flexible cell; pneumatic motors; knowledge base; expert system; flexible assembly cell; inference engine; simulated cell; gripper adaptor adjustments; vision system; cell operation

D

Title: Modeling Agv Systems.
Author: Davis, Deborah A.
Corporate Source: Systems Modeling Corp, State College, PA, USA
Conference Title: 1986 Winter Simulation Conference Proceedings.
Conference Location: Washington, DC, USA Conference Date: 1986 Dec 8-10
Source: Winter Simulation Conference Proceedings 1986. Publ by IEEE, New York, NY, USA. Available from IEEE Service Cent (Cat n 86CH2385-3), Piscataway, NJ, USA p 568-574
ISSN: 0275-0708 ISBN: 0-911801-11-1
Abstract: Computer simulation is often used as an analysis tool during the design of automated guided vehicle (AGV) systems. However, because of the complexities inherent in automated material handling systems, general-purpose simulation languages must be used creatively to capture the desired detail in the model. Some general concepts which can be used to model AGV systems are presented. Some of the critical concerns which must be addressed in a simulation analysis of an AGV system are discussed. 10 refs.
Descriptors: *Vehicles--*Automation; Computer Simulation; Computer Simulation Languages; Materials Handling--Automation
Identifiers: Automated Guided Vehicles; General-Purpose Simulation Languages; Outgoing Queue Size; Vehicle Staging; Modeling Vehicle Failure

Title: Evaluating AGVS Circuits By Simulation.
Author: Duffau, B.; Bardin, C.
Corporate Source: Seri Renault Automation, Fr
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 229-245
ISBN: 0-903608-98-7
Abstract: Firstly, the authors present the sizings and evaluations of AGVS circuits, using analytical, statistical and simulation approaches. Focusing particular attention on evaluation methods by simulation (after an analytical presizing, they present networks modelised by queue software and Petri nets. The authors then give a detailed description of the functions required by a specialised AGVS simulator. Finally, the authors present the results from research and experiments, and examples using such a modelling.
Descriptors: *VEHICLES--*Automation; COMPUTER SIMULATION
Identifiers: Automated Guided Vehicles Systems (AGVS); Specialised AGVS Simulator; Cleon Renault Workshop

Title: AGVS Design In Automotive Industry.
Author: Duffau, B.; Bloche, E.
Corporate Source: Renault Automation Seri, Fr
Conference Title: Proceedings of the 4th International Conference on Automated Guided Vehicle Systems: AGVS-4.
Conference Location: Chicago, IL, USA Conference Date: 1986 Jun 24-26
Source: Publ by IFS (Publ), Kempston, Engl p 35-49
ISBN: 0-938507-13-6

Abstract: How can an AGVS be designed that functions day and night's shift, capable of absorbing 30% loads fluctuations in a motor manufacturing workshop? In a first part the authors try to give an answer to this question. To do this they compare classical methods of dimensioning to simulation techniques. In a second part, they compare the constraints encountered in the automotive industry to other manufacturing units (distribution center, electronics workshop, assembly workshop etc. . .) in the designing of AGV network. The authors point out the problems that designers of AGVS have to deal with in designing and in defining piloting techniques. Finally a user-oriented simulation software is presented, that, in the vast majority of cases, allows the users to design AGV networks and test their characteristics. (Author abstract)

Descriptors: *Automobile Manufacture--*Automation; Vehicles--Industrial Applications; Computer Simulation

Identifiers: Motor Manufacturing Workshop; Automotive Industry; User-Oriented Simulation Software; Automated Guided Vehicle System (AGVS); AGV Networks

E

Title: Pull versus push strategy for automated guided vehicle load movement in a batch manufacturing system

Author(s): Egbelu, P.J.

Pennsylvania State Univ., University Park, PA, USA

J. MANUF. SYST. VOL. 6, NO. 3 pp. 209-222 1987

Abstract: Most automated guided vehicle (AGV) dispatching rules reported in literature to date are based on attributes of the vehicles and the workcenters from where the loads originate. These rules can therefore be described as source driven vehicle dispatching rules. In none of these rules is the load pickup priority determined based on the states of the load destinations. In this paper, an algorithm that assigns load movement priority based on the demand states of the load destinations is presented. Analysis indicates that the algorithm performs competitively with some of the best source driven rules currently reported. It also provides a material flow and workcenter priority assignment flexibility not matched by any source based rules.

Descriptors: computer aided manufacturing; automated guided vehicles; simulation; distribution

Title: Design methodology for operational control elements for automatic guided vehicle based material handling systems

Author(s): Egbelu, P.J.

Virginia Polytech. Inst. and State Univ., Blacksburg, VA, USA

DISS. ABST. INT. PT. B - SCI. + ENG. VOL. 44, NO. 1 1983

Pagination: 373 pp

Abstract: A methodology for the design of operational control aspects of an Automatic Guided Vehicle (AGV) based material handling system is presented. The methodology, which is composed of an integrated model of an AGV based system, was implemented using simulation technique. The model views a manufacturing function as consisting of machining, queueing, and moving of parts in a shop and that these components of manufacturing must be integrated and co-ordinated if the production objectives of an enterprise are to be realized. A machining center is modeled as a physical region of a plant and it consists of machines for part processing and capacitated queues in which inbound and outbound parts reside, queueing for machining of handling resources.

Descriptors: mechanical handling; guidance and control systems; automation; computer aided manufacturing; modeling; dissertation

Title: Concurrent specification of unit load sizes and automated guided vehicle fleet size in manufacturing system

Author: Egbelu, Pius J.

Corporate Source: Pennsylvania State Univ, University Park, PA, USA

Source: International Journal of Production Economics v 29 n 1 Feb 1993. p 49-64

ISSN: 0925-5273

Abstract: Even though many subproblems related to the design of automated guided vehicle (AGV) systems interact with one another, current design approaches of AGV systems tend to focus on solving one subproblem at a time. Piecemeal solution leads to suboptimal design. What is required is a concurrent engineering design approach that integrates two or more subproblems into a design model which, when solved, yields the optimal design parameters for the entire system. Three areas that are amenable to integrated design for AGV systems are unit load size specification, vehicle size selection, and fleet size determination. Even though the size and number of unit loads to be moved per period constitute major factors in the determination of vehicle requirements and the specification of vehicle size for AGV based systems, there is very little research done in the specification of load sizes to optimize on total vehicle requirements. In his paper, a procedure is presented for the selection of the best unit load sizes for all part types manufactured in a shop that employs AGVs for handling. The problem is modeled as a mathematical program and solved by a hybrid algorithm that includes numerical search, computer simulation, and statistical analysis. The decision criterion is based on the minimization of expected total manufacturing cost. (Author abstract) 29 Refs.

Descriptors: *Materials handling; Automation; Manufacture; Optimization; Vehicles; Algorithms; Statistical methods; Mathematical models; Costs

Identifiers: Automated guided vehicles; Vehicle size selection; Vehicle fleet size determination; Concurrent engineering; Total manufacturing costs

Title: Characterization Of Automatic Guided Vehicle Dispatching Rules.
Author: Egbelu, Pius J.; Tanchoco, Jose M. A.
Corporate Source: Syracuse Univ, Dep of Industrial Engineering & Operations Research, Syracuse, NY, USA
Source: International Journal of Production Research v 22 n 3 May-Jun 1984 p 359-374
ISSN: 0020-7543
Abstract: Hardware failures notwithstanding, the ability of an automated system operating according to promised potential is dependent upon the operational control measures in force. Some heuristic rules for dispatching Automatic Guided Vehicles (AGVs) in a job shop environment are presented. The rules are useful for assigning priorities to work stations requesting the services of a vehicle for material pickup. The likely effects of these rules on the performance of a job shop are postulated. Simulations results to demonstrate the effects of these rules are also presented. 18 refs.
Descriptors: Materials Handling--*Control; Computer Simulation; Industrial Plants--Automation
Identifiers: Automatic Guided Vehicles

Title: Designing The Operations Of Automatic Guided Vehicle System Using AGVSim.
Author: Egbelu, Pius J.; Tanchoco, Jose M. A.
Corporate Source: Syracuse Univ, Dep of Industrial Engineering & Operations Research, Syracuse, NY, USA
Conference Title: Proceedings of the 2nd International Conference on Automated Guided Vehicle Systems and 16th IPA Conference (Institute for Production and Automation).
Conference Location: Stuttgart, West Ger Conference Date: 1983 Jun 7-9
Source: Publ by IFS (Publ) Ltd, Kempston, Bedford, Engl and North-Holland Publ Co, Amsterdam, Neth and New York, NY, USA p 21-30
Descriptors: *INDUSTRIAL TRUCKS--*Computer Aided Design

Title: Potentials For Bi-Directional Guide-Path For Automated Guided Vehicle Based Systems.
Author: Egbelu, Pius J.; Tanchoco, J. M. A.
Corporate Source: Pennsylvania State Univ, University Park, PA, USA
Source: International Journal of Production Research v 24 n 5 Sep-Oct 1986 p 1075-1097
ISSN: 0020-7543
Abstract: Most current applications of automated guided vehicle systems (AGVS) in manufacturing shop environments employ uni-directional guide-paths for vehicle routing despite the fact that bi-directional vehicles exist. In this paper comparisons and issues regarding uni-directional and bi-directional flows are presented. Also presented is a model of a bi-directional traffic flow guide-path. The effect of the traffic flow pattern on the shop throughput is demonstrated and compared to that of a uni-directional flow system of an equivalent facility. The model is implemented using computer simulation. (Author abstract)
Descriptors: Vehicles--Guideways; Transportation--Traffic Control; Industrial Trucks--Automation
Identifiers: Automated Guided Vehicles; Bi-Directional Guide-Path

Title: Task Assignment Model for Automatic Guided Vehicle Based Material Handling Systems
Author(s): Egbelu, P.J.; Tanchoco, J.M.A.
VA Polytechnic Inst. & St. Univ.
Thirteenth Annual Pittsburgh Conference on Modeling and Simulation 8220193 Pittsburgh, PA 22-23 Apr 82
Department of Electrical Engineering, University of Pittsburgh
1982, Proceedings available: W.G.Vogt, Modeling and Simulation Conference, 348 Benedum Engineering Hall, University of Pittsburgh, Pittsburgh, PA 15261

F

Title: ROVER BY ROBOT.
Author: Fryatt, Arthur
Corporate Source: Manufacturing Systems, London, Engl
Source: Manufacturing Systems v 5 n 11 Nov 1987 p 16-17
ISSN: 0748-948X
Abstract: The use of computers during styling, design, engineering and production of automobiles by Austin Rover Group is described. Because the use of computer simulation allows ideas to be validated before prototypes are made, (CIE) facilities have effected shorter product cycle times. Automation of production techniques requires components to be more dimensionally consistent than previously. By using a common computer, shared by all designers and engineers, more work can be carried out in parallel. CIE integrates product design, tooling, production engineering and final manufacturing, ensuring that the end product is a faithful interpretation of the designer's original intent.
Descriptors: *Automobile Plants--*Flexible Manufacturing Systems; Robots, Industrial--Vision Systems; Automobiles--Computer Aided Design; Inspection --Automation; Computer Simulation--Industrial Applications Identifiers: Computer-Integrated Engineering (CIE); Integrated Technical Strategy; Computer Numerical Controlled (CNC); Automated Guided Vehicles (AGVs); Robotic Subassembly Cells

G

Title: AGVSim2 - a development tool for AGVS controller design.
Author: Gaskins, Robert J.; Tanchoco, J. M. A.
Corporate Source: Purdue Univ, West Lafayette, IN, USA
Source: International Journal of Production Research v 27 n 6 Jun 1989 p 915-926
ISSN: 0020-7543
Abstract: This paper presents a vehicle system simulator, AGVSim2, to be used in the evaluation of control strategies of real-time free-ranging vehicle supervisory controllers. The C-based discrete-event simulator is designed to be flexible so that many system configurations can be modelled. Also, the simulator allows for complex free-ranging vehicle manoeuvres, bidirectional flow, multiple vehicle types, and multiple loads on a vehicle. Instead of modelling the vehicle controller logic, the simulator is linked directly to the vehicle supervisory controller software. The simulator reports a variety of performance statistics and a trace of the system activities so that the performance of the supervisory controller can be evaluated. To improve the evaluation procedure, the simulator is also designed to be linked to an animation module that graphically displays the vehicle movements. (Author abstract) 27 Refs.
Descriptors: *Materials Handling--*Control Systems; Control Systems, Programmed--Computer Simulation; Control Systems--Design; Vehicles-- Automatic Guidance; Computer Systems Programming--Supervisory And Executive Programs
Identifiers: AGVs Controller Design; Automated Guided Vehicle System (AGVS); AGVS Simulator; Vehicle Supervisory Controller Software; Supervisory Controller Performance; Performance Statistics

Title: An AGVS simulation code generation for manufacturing applications.
Author: Gong, Dah-Chuan; McGinnis, Leon F.
Corporate Source: Sch of Ind & Syst Eng, Georgia Inst of Technol, Atlanta, GA, USA
Conference Title: 1990 Winter Simulation Conference Proceedings
Conference Location: New Orleans, LA, USA Conference Date: 1990 Dec 9-12
Source: 90 Winter Simulation Conf. Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 90CH2926-4). p 676-682
ISSN: 0275-0708 ISBN: 0-911801-72-3
Abstract: The concept of developing an automatic simulation method in connection with the design of an automated guided vehicle system (AGVS) is considered. A simulation code generator (SCG) implemented to demonstrate this concept is described. The generator converts input data, provided by the designer, into a SIMAN simulation program for evaluating a manufacturing system with automated guided vehicles moving along a unidirectional guidepath network. An operation scheme which describes the information flow in an AGVS is also presented to aid the understanding of the system logic and the development of the SCG. The simulation model logic is illustrated by a network diagram in which each node corresponds to an event or a process. A case study is used to demonstrate the SCG. 10 Refs.
Descriptors: *Computer Simulation--*Industrial Applications; Computer Aided Manufacturing
Identifiers: Automated Guided Vehicle System(AGVS); Simulation Modeling

H

Title: Simulation Model and Analysis; Integrating AGV's with non-automated Material Handling.
Author: Harmonosky, Catherine M.; Sadowski, Randall P.
Corporate Source: Purdue Univ, School of Industrial Engineering, West Lafayette, IN, USA
Conference Title: 1984 Winter Simulation Conference Proceedings.
Conference Location: Dallas, TX, USA Conference Date: 1984 Nov 28-30
Source: Winter Simulation Conference Proceedings 1984. , Publ by IEEE, New York, NY, USA, Available from IEEE Service Cent (Cat n 84CH2098-2), Piscataway, NJ, USA p 341-347
ISSN: 0275-0708 ISBN: 0-911801-04-9
Abstract: This study presents the logic for a general purpose simulation model representing an automatic guided vehicle (AGV) system integrated with traditional materal handling equipment. To accurately represent this type of system, logic includes vehicle loading/unloading and conventional equipment passing capability. The model's flexibility accommodates any combination of straight aisles and intersections through minor adjustments to the general model. The logic concepts are implemented using the SIMAN simulation language. An application is presented which demonstrates the use of the model to analyze the impact of interfacing AGV and traditional traffic upon aisle congestion and overall system performance. 8 refs.
Descriptors: *Materials Handling--*Mechanization; Computer Simulation; Vehicles--Control; Logic Circuits
Identifiers: AUtomatic Guided Vehicles; Simulation Tools; Simulation Language Siman

Title: AGV (Automated Guided Vehicle) applications
Author(s): Hopkins, D.
GE Aircr. Eng.
2nd International Symposium on Robotics and Manufacturing 8845002 Albuquerque, NM USA 16-18 Nov 1988
College of Engineering, University of New Maxico CAD Lab. Systems/Robotics, EECE Department, University of New Maxico, Albuquerque, NM 87131 USA
ISBN: 0-7918-0006-7. Price: $90.00 + Mailing Cost

Title: Use Of Physical Model Simulation To Emulate An AGV Material Handling System.
Author: Hurley, R. G.; Coffman, P. E.; Dixon, J. R.; Walacavage, J. G.
Corporate Source: Ford Motor Co, Dearborn, MI, USA
Conference Title: Proceedings - 1987 IEEE International Conference on Robotics and Automation.
Conference Location: Raleigh, NC, USA Conference Date: 1987 Mar 31-Apr 3
Source: Publ by IEEE, New York, NY, USA. Available from IEEE Service Cent (Cat n 87CH2413-3), Piscataway, NJ, USA p 1040-1045
ISBN: 0-8186-0787-4
Abstract: An application of physical modeling to the simulation of a prototype AGV (automatic guided vehicle) material handling system is considered. Physical modeling is the study of complex automated manufacturing and material handling systems through the use of small scale components controlled by mini- and/or microcomputers. By modeling the mechanical operations of the proposed AGV material handling systems, it was determined that control algorithms and AGV dispatch rules could be developed and evaluated. The authors presents an explanation of physical modeling as a simulation tool and address the development of the control algorithm, dispatching rules, and a prototype physical model of a flexible machining system (FMS). Quantitative results obtained in the operation and evaluation of the AGV dispatching rules are reported. 5 refs.
Descriptors: *Materials Handling--*Robot Applications; Robots, Industrial --Mobile; Industrial Plants--Flexible Manufacturing Systems; Process Control--Computer Applications
Identifiers: Automatic Guided Vehicle; AGV Material Handling System; Dispatching Rules; Flexible Machining System; Physical Model Simulation

I

Title: Multi-layer expression and application of PLUS/F simulation model - description of a production system with AGVs.
Author: Inoue, Ichiro; Fuyuki, Masahiko; Kida, Tomoyuki; Nakahira, Tadashi; Ito, Ikuo
Corporate Source: NEC Corp, Kawasaki, Jpn
Source: Bulletin of the Japan Society of Precision Engineering v 22 n 4 Dec 1988 p 283-288
ISSN: 0582-4206
Abstract: A multi-layer expression method is proposed to enhance simulation model descriptiveness for a production system with AGVs (Automated Guided Vehicles). The production system with AGVs is viewed as a composite system which consists of a material flow subsystem and a transportation subsystem. In a multi-layer expression, a simulation model of each subsystem is expressed on one layer, and a superposition of layers forms a frame of the total system model expression. to concretize the representation of each subsystem, the PLUS/F simulation model is employed. Cooperative and cross-reference functions which represent modes of subsystem coupling are introduced s extensions of the original model. Further, a practical application of the model to an actual case is demonstrated. (Edited author abstract) 6 Refs.
Descriptors: *Production Engineering--*Simulation; Vehicles--Guideways; Robots, Industrial--Mobile; Materials Handling; Printed Circuits-- Manufacture
Identifiers: Automated Guided Vehicles; Multilayer Expression Method; Plus/F Simulation Model; Printed Circuit Board Assembly Shop

J

Title: Hierarchical Approach To Computer Animation In Simulation Modeling.
Author: Johnson, M. Eric; Poorte, Jacob P.
Corporate Source: Systems Modeling Corp, State College, PA, USA
Source: Simulation v 50 n 1 Jan 1988 p 30-36
ISSN: 0037-5497
Abstract: The trend toward animation in simulation yields interesting and exciting new techniques for simulation modeling. We discuss some of the benefits of using animation and propose a hierarchical aproach to effectively use animation during all phases of the modeling process. We then illustrate the use of this hierarchical structure with a manufacturing case study involving the integration of AGV's with non-automated material handling devices. (Author abstract) 15 refs.
Descriptors: Computer Aided Manufacturing; Computer Simulation; Computer Graphics--Animation; Systems Science And Cybernetics--Hierarchical Systems; Materials Handling
Identifiers: DISCRETE-EVENT SIMULATION

K

Title: AGV System Simulation - A Tool For Better Design.
Author: Karlstrom, Hugo
Corporate Source: SattControl UK Ltd
Conference Title: Colloquium on Simulation as an Aid to Plant Design.
Conference Location: London, Engl Conference Date: 1986 Mar 5
Source: IEE Colloquium (Digest) n 1986/31. Publ by IEE, London, Engl p 4. 1-4. 6
Abstract: Automated Guided Vehicles (AGVs) are more and more commonly being used for the transport of goods in factories and warehouses. The planning of an AGV system is extremely difficult to do by manual calculations, because of the many variables which define the system. This paper presents an AGV system simulation package, which makes it possible to run and check the AGV system long before it is installed on site. The AGV system simulation package uses advanced computer graphics on a personal computer. The simulation shows, on the VDU screen, AGVs moving around the layout. Performance statistics can be accessed at any time during the simulation. (Author abstract)
Descriptors: *Computer Simulation--*Applications; Materials Handling; Computer Graphics; Computers, Personal
Identifiers: Automated Guided Vehicles; Performance Statistics

Title: Optimal scheduling for an automated two-machine manufacturing system
Author(s): Kise, H.; Kohno, K.; Shioyama, T.; Kushiyama, T.
Trans. Japan Soc. Mech. Eng. (Ser. C) Vol. 57, No. 537 Pp. 1776-1782 1991
Abstract: This paper discusses a scheduling problem of minimizing the maximum completion time for an automated manufacturing system such as FMS and FMC that consists of two automated machining cells such as machining centers, an AGV (automated guided vehicle) and loading and unloading stations. This paper develops a branch-and-bound algorithm and an approximation algorithm for this problem. Computer simulations implemented show that the branch-and-bound algorithm can solve most the examples tested up to 200 jobs in a widd range of the problem, and the approximation algorithm can yield schedules with relative errors smaller than 1(() at worst.
Descriptors: flexible manufacturing systems; automated guided vehicles; production scheduling

Title: AGVs Help To Raise Productivity In Assembly By Organisational Means.
Author: Koether, Reinhard
Corporate Source: Fraunhofer-Inst fuer Arbeitswirtschaft und Organisation, Stuttgart, West Ger Conference Title: Proceedings of the 2nd International Conference on Automated Guided Vehicle Systems and 16th IPA Conference (Institute for Production and Automation).
Conference Location: Stuttgart, West Ger Conference Date: 1983 Jun 7-9
Source: Publ by IFS (Publ) Ltd, Kempston, Bedford, Engl and North-Holland Publ Co, Amsterdam, Neth and New York, NY, USA p 289-300
ISBN: 0-903608-45-6
Descriptors: *Industrial Trucks--*Computer Simulation Identifiers: Automated Guided Vehicle Systems; Line Assembly System ; Cost Reduction; Evaluation Analysis; Work Structuring; Labor Use; Production Rate; Work Distribution

Title: AGVs For Highly Productive Assembly Systems-A Sophisticated Simulation Tool To Optimize The System Design.
Author: Koether, R.; Letters, F.
Corporate Source: Fraunhofer Inst of Industrial Engineering, West Ger
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 317-334
ISBN: 0-903608-98-7
Abstract: With the flexible transportation system AGVS flexible and highly productive assembly systems can be realized. To gain improved results of the systems design, simulation can be a powerfull tool. A simulation system is presented, that allows dynamic simulation not only of the AGVS but also of the assembly system, its organization and the production program. The simulation results can be used to optimize the planned system and to evaluate the preferability of the planned assembly system with AGVS. (Author abstract) 17 refs.
Descriptors: *Assembly Machines--*Automation; Vehicles--Automation; Productivity
Identifiers: Automated Guided Vehicle Systems (AGVS); Flexible Transportation System; Assembly System Productivity; Assembly Model; Production System

Title: Heuristic unidirectional flowpath design approaches for automated guided vehicle systems.
Author: Kouvelis, Panagiotis; Gutierrez, Genaro J.; Chiang, Wen-Chyuan
Corporate Source: Duke Univ, Durham, NC, USA
Source: International Journal of Production Research v 30 n 6 Jun 1992 p 1327-1351
ISSN: 0020-7543
Abstract: In this paper, we address the flowpath design issue of automated guided vehicle systems (AGVSs). In particular we concentrate on the design of unidirectional flowpaths (i.e. vehicles are restricted to travel only in one direction along a given segment of the flowpath). We have developed five different heuristics for the design of unidirectional AGVSs. We have also developed simulated annealing algorithms for the above problem. Our extensive computational results indicate that a composite heuristic (i.e. one that combines the most successful of our five heuristics) yields solutions of comparable quality in a fraction of the time required by simulated annealing. For large-size flowpath design problems, we advocate the usage of composite heuristics over simulated annealing, and in particular for the cases in which inaccurate estimates for the input data in our design problem exist. (Author abstract) 35 Refs.
Descriptors: *Vehicles--*Guideways; Materials Handling--Automation; Mathematical Techniques--Heuristic; Management--Information Systems; Computer Networks
Identifiers: Unidirectional Flowpath Design; Automated Guided Vehicles Systems; Simulated Annealing; Composite Heuristic

Title: General Edp-Aided Planning And Realization Of Agv-Systems.
Author: Kuhn, A.; Schmidt, F.
Corporate Source: Fraunhofer-Inst fuer Transporttechnik und Warendistribution, West Ger
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 247-258
ISBN: 0-903608-98-7
Abstract: Extensive efforts on AGVS-planning, above all drawing up several alternatives, optimizing, evaluating and selection, more and more justifies computer-aid, in order to achieve cost effective work at a highlyqualified level. For this purpose software-tools as components of a general EDP-aided planning-instrument for AGV-systems have been developed at the Fraunhofer-Institute in Dortmund. This report outlines the possibilities of how to use the computer during the comprehensive planning process of AGVS-facilities, including each step from processing the initial data up to the simulation and final realization. The developed, individual components of the project will be explained with the help of selected examples. (Author abstract) 4 refs.
Descriptors: *Vehicles--*Automation; Computer Simulation; Computer Software
Identifiers: Automated Guided Vehicles Systems (Agvs); Software-Tools; Software-Tools; Systemload Analysis; Agvs-Planning Approach

L

Title: AGVs: The Latest In Material Handling Technology.
Author: Lasecki, Robert R.
Source: CIM Technol v 5 n 4 Winter 1986 p 90-94
Abstract: Material handling techniques have developed only slightly over the past decade. The primary growth in the technology has been in the areas of robotics and automated guided vehicle (AGV) systems. As processes and processing methods were developed, new manipulation and control techniques were implemented, providing automated machining centers, flexible manufacturing systems, and robotic workstations. These areas were refined a good deal before significant consideration was given to the movement of product and base material among receiving, storage, and interim points of use or individual workcenters. After-the-fact implementation of a material transportation system often resulted in poor performance. Advances in automated guided vehicles are described in the article.
Descriptors: Materials Handling--Equipment; Vehicles--Automation; Computers--Applications; Computer Simulation; Accident Prevention; Ultrasonic Applications
Identifiers: Automated Guided Vehicle; On-Board Vehicle Programming; Intervehicle Blocking; Tow Tractor; Computer Blocking

Title: Evaluation of automated guided vehicle systems by simulation.
Author: Lee, Jim; Choi, Richard Hoo-Gon; Khaksar, Majid
Corporate Source: Univ of Southwestern Louisiana, Lafayette, LA, USA
Conference Title: Proceedings of the 12th Annual Conference on Computers and Industrial Engineering
Conference Location: Orlando, FL, USA Conference Date: 1990 Mar 12-14
Source: Computers & Industrial Engineering v 19 n 1-4 1990. Publ by Pergamon Press Inc, Elmsford, NY, USA. p 318-321
ISSN: 0360-8352
Abstract: An experimental study was conducted in this research to evaluate the design of Automated Guided Vehicle (AGV) systems using discrete event simulation. Given the layout of work stations and several types of jobs to be processed in the system, the constituent elements of the AGV system, including the number of AGVs, the speed of the vehicles, the number of machines at each station, the number of load and unload stations, rules for selection of AGV, routing criteria, and track intersection priorities, were defined in a SIMAN model. Pilot runs were performed first to determine the values of the system parameters which may affect the performance of the AGV systems. A factorial design was then used to generate simulation experiments. Finally, a number of simulation runs were executed under different experimental conditions to obtain preliminary statistics on the following performance measures utilization of AGVs, average number of jobs waiting in the queue for an AGV, average system throughput time, average waiting time in intersections and total waiting time in queues. (Author abstract) 3 Refs.
Descriptors: Vehicles--Guideways; Computer Simulation; Machine Shops-- Layout; Materials Handling--Costs; Statistical Methods
Identifiers: Automated Guided Vehicle (AGV); AGV Systems Simulation; AGV Track Layout; Route Analysis

M

Title: Simulation In The Design Of Paint-Line Control Logic.
Author: Madsen, William C.
Corporate Source: John Deere Harvester Works, East Moline, IL, USA
Conference Title: Proceedings of the Third Annual Control Engineering Conference. (Held as part of the Control Engineering Conference & Exposition.)
Conference Location: Rosemont, IL, USA Conference Date: 1984 May 22-24
Source: Proceedings of the Annual Control Engineering Conference 3rd. Publ by Control Engineering, Barrington, IL, USA p 315-322
Abstract: Paint systems are one of the more expensive monuments in manufacturing facilities. Design of these systems is crucial in making effective use of these facilities. Through simulation, two areas of design were investigated. First, the simulation was used to design the conveyor and its controls. Second, it was used to define the logic to control the input sequence of unpainted material. In meeting the objective, simulation demonstrated the necessity of integrating these historically segregated aspects of manufacturing systems into one network.
Descriptors: Control Systems- design; Conveyors-Control; Process Control- Computer Applications
Identifiers: Control Hardware; Paint-Line Control Logic; Automatic Guided Vehicle Systems

Title: Bridging the gap: Transferring logic from a simulation into an actual system controller.
Author: McHaney, Roger
Corporate Source: Control Engineering Co, Harbor Springs, MI, USA
Conference Title: Winter Simulation Conference Proceedings - 1988
Conference Location: San Diego, CA, USA Conference Date: 1988 Dec 12-14
Source: Winter Simul Conf Proc 1988 Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat n 88CH25122-2), Piscataway, NJ, USA. p 583-590
Abstract: The author demonstrates the importance of maintaining simulation integrity during actual system implementation and presents four methods of logic transfer to aid in accomplishing this: philosophic transfer, pseudocode transfer, database transfer, and actual code transfer. The four methods are discussed and compared using the example of an automatic guided vehicle system simulation. 14 refs.
Descriptors: *Computer Simulation--*Applications; Control Systems; Vehicles; Logic Design; Database Systems
Identifiers: Automatic Guided Vehicles; Pseudocode Transfer; Philosophic Transfer

Title: Design of an automated guided vehicle-based material handling system for a flexible manufacturing system

Author(s): Mahadevan, B.; Narendran, T.T.
Dep. Hum. and Soc. Sci., Indian Inst. Technol., Madras 600036, India

INT. J. PROD. RES. VOL. 28, NO. 9 pp. 1611-1622 1990

Abstract: Automated guided vehicle (AGV)-based material handling systems (MHSs), which are widely used in several flexible manufacturing system (FMS) installations, require a number of decisions to be made. These include the number of vehicles required the track layout, traffic pattern along the AGV tracks, and solving traffic control problems. This paper addresses the key issues involved in the design and operation of AGV-based material handling systems for an FMS. The problems arising from multi-vehicle systems are analysed, and strategies for resolving them are examined using analytical and simulation models.

Descriptors: automated guided vehicles; flexible manufacturing systems; mechanical handling; automated guided vehicles; production design; production costs

Title: Use of Simulation - From Specification of an FMS Through to Its Operation - A Case Study.

Author: Makin, A. J.
Corporate Source: Leyland Trucks Ltd, Engl

Conference Title: Proceedings of the 2nd International Conference on Simulation in Manufacturing: SIM-2.
Conference Location: Chicago, IL, USA Conference Date: 1986 Jun 24-26
Source: Publ by IFS (Conferences) Ltd, Kempston, Engl p 249-256
ISBN: 0-948507-14-4

Abstract: A flexible manufacturing system (FMS), manufacturing automotive components, was installed at the Leyland Bus gearbox plant in Lancashire, England, during 1985. Its AGV-based material handling system serves single spindle machining centres, all of which are operated by a sophisticated, central computer control system, allowing 24 hour/day operation with 8 hours of unmanned production. This paper describes the use of various simulation models over a four year period, from initial specification, followed by functional design, through to the use of a production planning model to evaluate alternative operating strategies during the first year of FMS operation. Particular reference is made to the effect of simulation work upon control system design decisions taken, all FMS control software being developed specially for this system. (Edited author abstract)

Descriptors: *Production Control--*Computer Simulation; Industrial Plants --Flexible Manufacturing Systems; Systems Science And Cybernetics--Man Machine Systems; Materials Handling--Automation
Identifiers: FMS; System Design Decisions

Title: A Decision Support System For Automated Guided Vehicle System Design
Author(s): Malmborg, C.J.
Dep. Decis. Sci. and Eng. Syst., Rensselaer Polytech. Inst., Troy, NY 12180-3590, USA
APPL. MATH. MODEL. VOL. 16, NO. 4 pp. 170-180 1992
Abstract: An interactive decision support system (DSS) for automated guided vehicle (AGV) system design is described. The DSS allows the user to flexibly access an analytical model relating changes in the levels of design variables to performance measures and operating dynamics for control zone AGV systems. Using the DSS, the system designer can interactively screen preliminary design solutions prior to the development of the simulation models used to develop and validate detail design specifications. This makes it possible to explore a broader range of the design solution space in a process of intelligent enumeration. A sample problem is examined using the DSS.
Descriptors: decision support systems; automated guided vehicles; design

Title: A model for the design of zone control automated guided vehicle systems
Author(s): Malmborg, C.J.
Dep. Decis. Sci. and Eng. Syst., Rensselaer Polytech. Inst., Troy, NY 12180-3590, USA
INT. J. PROD. RES. VOL. 28, NO. 10 pp. 1741-1758 1990
An analytical modelling strategy is proposed as a screening device for use prior to the simulation phase in the design of zone control automated guided vehicle systems (AGVS). The proposed model predicts the effects of the major AGVS design variables on system-performance measures including maximum throughput capacity and risk factors associated with shop locking. This is done through a process which specifically evaluates the system operating dynamics associated with vehicle fleet size, guidepath layout, workstation storage capacity, routeing, and, to a less precise extent, vehicle dispatching. The objective of the analytical model is to reduce the extent of simulation modelling necessary to design a zone-control AGVS for a given materials-handling workload.
Descriptors: automated guided vehicles; design; zone control; routing

Title: Simulation evaluation of model based AGVS fleet sizing.
Author: Malmborg, Charles J.
Corporate Source: Rensselaer Polytechnic Inst, Troy, NY, USA
Conference Title: Proceedings of the 1990 International Industrial Engineering Conference
Conference Location: San Francisco, CA, USA Conference Date: 1990 May 20-23
Source: Proc 1990 Int Ind Eng International Industrial Engineering Conference Proceedings. Publ by IIE, Norcross, GA, USA. p 205-211
ISBN: 0-89806-112-1
Abstract: A test problem is used to compare AGVS fleet sizing results from a control zone based modeling procedure with those obtained using simulation. Consistent with previous studies comparing analytical and simulation models, the results suggest that analytical procedures tend to underestimate vehicle requirements. (Author abstract) 3 Refs.
Descriptors: *Vehicles--*Scheduling; Materials Handling--Automation; Computer Simulation; Control Systems--Mathematical Models; Probability
Identifiers: Automatic Guided Vehicle Systems; Fleet Sizing; Control Zone Models; System Design

Title: Engineering Workstation For Computer Aided Engineering Of Material Handling Systems.
Author: McGinnis, L. F.; Goetschalckx, M.
Corporate Source: Georgia Inst of Technology, Atlanta, GA, USA
Conference Title: Proceedings of Manufacturing International '88. v 1, Symposium on Product and Process Design; v 2, Symposium on Management and Economics; v 3, Symposium on Manufacturing Systems; v 4, Manufacturing Science of Composites.
Conference Location: Atlanta, GA, USA Conference Date: 1988 Apr 17-20
Source: Publ by ASME, New York, NY, USA v III, p 137-142
Abstract: Automated Guided Vehicle Systems (AGVS) are becoming more and more the main material handling devices in flexible manufacturing systems. The design of an AGVS network is a complex, iterative process. A Computer Aided Engineering (CAE) design tool must provide the following three capabilities: first, it must allow the easy, interactive and graphical definition of the system by providing specialized drawing symbols and specialized operations on those symbols. Second, it must support direct, interactive analysis of the design from within the design package. Third, it must provide the capability to interface with off-line simulation, optimization and other analysis programs. A prototype of such a CAE tool has been developed for the design of AGVS and has been implemented on a microcomputer. Development strategy, system structure, implementation issues, as well as future research areas, are reported on. (Edited author abstract) 6 refs.
Descriptors: *Materials Handling--*Computer Aided Engineering; Industrial Plants--Flexible Manufacturing Systems; Computer Interfaces; Mechanical Engineering--Design Aids; Computer Systems, Digital--Interactive Operation; Computer Simulation
Identifiers: Engineering Workstation; Automated Guided Vehicle Systems; Graphical User Interface; CAE; FMS

Title: Evolutions In The AGV Systems Supplied By A Materials Handling Manufacturer To Automobile Assembly Plants.
Author: Montalenti, U.; Malamo, C.; Miotto, G.; Ponsetto, M.
Corporate Source: Gruppo Pianelli & Traversa, Italy
Conference Title: Proceedings of the 4th International Conference on Automated Guided Vehicle Systems: AGVS-4.
Conference Location: Chicago, IL, USA Conference Date: 1986 Jun 24-26
Source: Publ by IFS (Publ), Kempston, Engl p 275-289
ISBN: 0-938507-13-6
Abstract: This paper discusses applications of automated guided vehicle systems (AGVS) for materials distribution and assembly lines feeding. AGVS as system integration in the automobile assembly, and in Flexible Manufacturing Systems (FMS) are described. Cases presented in this paper show that in more than 20 years' experence, a Company's AGVs technology has gradually developed to answer and sometime anticipate the evolving industrial needs. Among the improvements contributing to this end in the various succeeding generations, the authors recall the increased kinematic precision and operational capabilities of the vehicles, the techniques aimed at containing their number in a plant, like an accurate simulation and the 'in route' battery charge, and, mainly, their ability to dialogue in real time with the superior levels of computerized control systems.
Descriptors: Materials Handling--Automation; Assembly Machines-- Automation; Vehicles; Automobile Manufacture
Identifiers: Automated Guided Vehicle Systems (AGVS); AGVS Tractors; Assembly Line Feeding; Materials Distribution

Title: On-line AGV system planning
Author(s): Moreno, L.; Salichs, M.A.; Aracil, R.; Campoy, P.
Univ. Politech. Madrid, Spain
19th International Symposium on Automotive Technology and Automation 8840465 Monte Carlo (Monaco) 24-28 Oct 1988
Allen-Bradley
Allied Automatiom Ltd., 42 Lloyd Park Avenue, Croydon CR0 5SB (UK). Telephone: 01 681 3069. Telefax: 01 686 1490., ISBN: 0947719229
Descriptors: Electrical Engineering; Civil And Mechanical Engineering

N

Title: Automod/Autogram: A Collection Of Software Tools For Simulation Studies Of Industrial Systems.
Author: Norman, T. A.; Norman, V. B.
Corporate Source: Brigham Young Univ, UT, USA
Conference Title: Proceedings of the 1st International Conference on Simulation in Manufacturing.
Conference Location: Stratford-upon-Avon, Engl Conference Date: 1985 Mar 5-7
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 111-127
ISBN: 0-903608-84-7
Abstract: AutoMod is a collection of software tools for the simulation of industrial systems involving discrete unit material handling components such as Automated Guided Vehicles, Automated Storage and Retrieval Systems, pallet conveyors, towlines, transfer lines, monorails, overhead power and free conveyor, etc. AutoGram works with AutoMod to provide animated three dimensional displays of simulated systems and to provide a quick means of digitizing the geometry of the systems. This paper gives an overview of the AutoMod system description language. Included are examples of pallet conveyor and AGV models. (Author abstract)
Descriptors: *Systems Engineering--*Computer Simulation; Materials Handling--Automation; Computer Graphics--Three Dimensional Graphics; Computer Simulation Languages
Identifiers: Systems Geometry; Automated Guided Vehicles; Pallet Conveyors

Title: Design Of A Simulation Package For Automated Guided Vehicle Systems.
Author: Norman, Susan K.; Scheck, Donald E.
Corporate Source: Univ of Wyoming, Laramie, WY, USA
Source: Computers & Industrial Engineering v 11 n 1-4 1986, Proc of the 8th Annu Conf on Comput and Ind Eng, Orlando, FL, USA, Mar 19-21 1986 p 401-405
ISSN: 0360-8352

Abstract: The lack of analytical techniques in designing optimal network configurations and control policies in Automated Guided Vehicle Systems (AGVS) leaves simulation as the designer's primary tool. ASP is a general purpose AGVS Simulation Package developed in SIMAN and FORTRAN. This interactive package is designed to relieve the user of programming the simulation model and allow the user to input different design configurations quickly. The package consists of an interactive user interface, a model file which contains the generic portion of the system model and an experiment file which contains user defined parameters which are problem specific. The user enters the network topology, vehicle operating characteristics and selects system and simulation control policies. Waiting time at each intersection and station and total system time are the general output of the package and may be further analyzed using the SIMAN Output Processor. (Author abstract) 6 refs.

Descriptors: *Industrial Trucks--*Control; Materials Handling--Computer Simulation; Transportation--Operations Research

Identifiers: Automated Guided Vehicle Systems; Interactive Software Package

Title: Simulation Of Automatic Guided Vehicle Control Systems.

Author: Norman, Van B.

Corporate Source: AutoSimulations Inc, Bountiful, UT, USA

Conference Title: Proceedings of the Third Annual Control Engineering Conference. (Held as part of the Control Engineering Conference & Exposition.)

Conference Location: Rosemont, IL, USA Conference Date: 1984 May 22-24

Source: Proceedings of the Annual Control Engineering Conference 3rd. Publ by Control Engineering, Barrington, IL, USA p 311-314

Abstract: Automatic Guided Vehicles (AGVs) are a major component of many Flexible Manufacturing Systems (FMS) and other highly automated industrial processes. Computer simulation is a cost effective design tool for the evaluation and specification of AGV control systems. The performance of such vehicle systems is dependent on many factors. One such factor is the dispatch rule to assign vehicles to movement requests. Various rules are presented and the considerations for rule use and implementation are discussed.

Descriptors: *Control Systems--*Computer Applications; Process Control-- Computer Applications

Identifiers: Traffic Factor; Load Factor; Automatic Guided Vehicles (AGV) ; Flexible Manufacturing Systems (FMS)

O

Title: Analysis of the AGV loading capacity in a JIT environment
Author: Occena, Luis G.; Yokota, Toshiya
Corporate Source: Univ of Missouri-Columbia, Columbia, MO, USA
Source: Journal of Manufacturing Systems v 12 n 1 1993. p 24-35
ISSN: 0278-6125
Abstract: The problem of transport performance is inherent in just-in-time (JIT) systems where lot sizes are smaller. When the lot size is made smaller to keep inventory low, the number of delivery trips to replenish materials increases, which affects the transport performance. This situation will intensify with increased throughput requirements, as more parts move through the system in the same period of time. Adapting an automated guided vehicle (AGV) to carry multiple loads is an appealing alternative to increasing the number of single-load AGVs in a JIT system. A multiple-load AGV (MLAGV) will not increase traffic congestion, but will require a more sophisticated AGV system (AGVS) model to ensure just-in-time delivery. This paper extends the authors' previous work on modeling an AGVS in a JIT environment. It examines the alternative of using a MLAGV to meet higher throughput requirements in a JIT manufacturing system. Computer simulation was used to evaluate the effects of multiple loads on both transport and logistic performance. The results provide a useful guide in the design of MLAGV systems, particularly in manufacturing environments with low inventory and high throughput requirements. (Author abstract) 14 Refs.
Descriptors: *Just in time production; Ground vehicles; Control systems; Materials handling; Computer simulation; Scheduling; Inventory control; Electronic guidance systems; Raw materials; Transportation Identifiers: Automated guided vehicle (AGV); Multiple load automated guided vehicle; Vehicle dispatching; Work in process (WIP) inventories

Title: Analysis of the dispatching and loading capacity of automated guided vehicles in a just-in-time environment.
Author: Occena, Luis G.; Yokota, Toshiya
Corporate Source: Univ of Missouri, Columbia, MO, USA
Conference Title: Proceedings of the 1992 Japan - USA Symposium on Flexible Automation. Part 2 (of 2)
Conference Location: San Francisco, CA, USA Conference Date: 1992 Jul 13-15
Source: Flexible Automation 1992 Proc 92 Jpn USA Symp Flexible Autom. Publ by ASME, New York, NY, USA. p 1123-1126
Abstract: This paper examines the modelling of an automated guided vehicle system (AGVS) under a just-in-time (JIT) environment. The influence of a 'JIT perspective' is emphasized throughout the model. A dispatching rule for automated guided vehicles (AGVs) in a JIT system is developed. Comparisons with previously developed dispatching rules are made based on logistic and transport performance by applying computer simulation. Also, the effect of loading capacity on performance for a multiple load AGV (MLAGV) in a JIT system is examined. The results of this study provide a helpful guide in the design of a MLAGV system to handle a manufacturing environment with high throughput and low inventory requirements. (Author abstract) 7 Refs.
Descriptors: *Just In Time Production; Vehicles; Production Engineering Identifiers: Automated Guided Vehicles (AGV); Just-In-Time Environment; AGV Transport System; Multiple Load AGV Systems; Maximum Demand (MD) Dispatching Rule; AGV Logistic Performance

Title: AGV case study. Instrument panel assembly at Ford Motor Company's Kansas City Assembly Plant.
Author: Olson, Mathew K.
Corporate Source: Ford Motor Co, Dearborn, MI, USA
Source: SME Technical Paper (Series) MS 1989 var paging MS89-522
ISSN: 0161-6382
Abstract: An automatic guided vehicle system was successfully launched at Ford Motor Company's Kansas City Assembly Plant for instrument panel assembly in January of 1989. Since its launch the system has met 100% of all production goals. Successful implementation of an automatic guided vehicle system doesn't just happen, it requires detailed planning, research, simulation verification, hardware prove out and, most importantly, full commitment from the final system user. This paper details the technical approach, decisions and events that lead to the successful launch of the Ford Motor Kansas City Instrument Panel Assembly Automatic Guided Vehicle System. (Author abstract)
Descriptors: *Vehicles--*Control Systems; Computer Simulation; Materials Handling; Transportation--Automation
Identifiers: Automatic Guided Vehicle Systems; Instrument Panel Assembly

P

Title: Simulation Tools For Investigation And Design Of FMS Transportation Systems.
Author: Pavlovsky, V. E.; Kiril'chenko, A. A.; Prudkovsky, S. G.; Romanov, V. A.; Barbashova, T. F.
Corporate Source: Acad of Sciences of the USSR, Moscow, USSR
Conference Title: Information Control Problems in Manufacturing Technology 1986, Proceedings of the 5th IFAC/IFIP/IMACS/IFORS Conference.
Conference Location: Suzdal, USSR Conference Date: 1986 Apr 22-25
Source: IFAC Proceedings Series n 1 1987. Publ by Pergamon Press, Oxford, Engl and New York, NY, USA p 105-109
ISSN: 0741-1146 ISBN: 0-08-033469-5
Abstract: The software developed for simulation and optimization of the flexible manufacturing systems (FMS) transportation systems is described. It allows solving three interrelated problems: (1) choice and arrangement of equipment in the FMS workstation; (2) simulation and evaluation of the FMS transportation subsystem operation; and (3) development of efficient algorithms for constructing the automated guided vehicle (AGV) route. The appropriate mathematical models and the numerical experiment results are presented. The software developed is an integral part of a programming complex to allow the computer-aided design and synthesis of the control modes for the FMS subsystems. (Author abstract) 4 refs.
Descriptors: *Materials Handling--*Computer Simulation; Robots, Industrial--Materials Handling Applications; Systems Engineering--Computer Aided Design
Identifiers: Fms; Cad; Automated Guided Vehicles

Title: Socio-Technique or Experiences and Possibilities Through Adoption of "Automated Guided Vehicle Systems for Increased Productivity, Within Volvo"
Author(s): Persson, I.
AB Volvo, Autocarrier Sys. (ACS)
1981 Spring Annual Conference and World Productivity Congress. 8120316 Detroit,MI.17May 81
American Institute of Industrial Engineers, Inc.
AIIE Catalog No. P-254
Descriptors: General Engineering and Technology; Civil and Mechanical Engineering
Section Heading: General Engineering And Technology ; Multidisciplinary ; Civil And Mechanical Engineering

Title: Simulating a factory production process with automated guided vehicles
Author(s): Petkovska, G.; Nedeljkovic, V.; Hovanec, M.; Puente, E.A.; Nemes, L. (eds.)
Lola Inst., Beograd, Yugoslavia
Information Control Problems In Manufacturing Technology. Pp. 455-460 1990
Publ: Pergamon Press, Inc., Maxwell House, Fairview Park, Elmsford, Ny 10523 (Usa),
Abstract: Paper presents a simulation model of a workshop (factory) for the production of multi-layer printed circuit boards, which represents a flexible manufacturing system (FMS). Several automated guided vehicles perform the material handling. The second version of GPSS-F simulation package was used for this simulation.
Descriptors: simulation; factory automation; flexible manufacturing systems; production control; mechanical handling; automated guided vehicles ; electronic equipment manufacture; printed circuit boards

Analysis of different AGV control systems in an integrated IC manufacturing facility, using computer simulation
Author(s): Prasad, K.; Rangaswami, M.; Abrams, M.A.; Haigh, P.L.; Comfort, J.C. (eds.)
Intel Corp., MS CH2-31, 5000 W Chandler Blvd., Chandler, AZ 85226, USA
1988 WINTER SIMULATION CONFERENCE. pp. 568-574 1989
1988 Winter Simulation Conference 12-14 Dec 1988 San Diego, CA (USA)
Publ: Society For Computer Simulation International, P.O. Box 17900, San Diego, Ca 92117 (Usa),
Abstract: In an integrated Manufacturing Facility, consisting of various individual manufacturing cells, an automated material handling system (AMHS) - such as an Automated Guided Vehicle System (AGVS) - servicing each of cells plays a major role. In order to understand, and choose, the correct type of AGV system, simulation techniques can be used very effectively. In this paper, two different AGV Control Systems have been analyzed with computer simulation models. These animated graphic models also include details of all the individual process steps from the Water Sort operation through the Test processes of an IC Manufacturing Facility at Intel.
Descriptors: computer integrated manufacturing; simulation; control systems; automated guided vehicles; electronic equipment manufacture; production control

Q

Title: Application of Simulation Modeling in Design of AGV System.
Author: Quinn, Edward B.
Corporate Source: AutoSimulations Inc, Bountiful, UT, USA
Conference Title: Proceedings of the ISA/87 International Conference and Exhibit.
Conference Location: Anaheim, CA, USA Conference Date: 1987
Source: Advances in Instrumentation v 42 pt 3. Publ by ISA, Research Triangle Park, NC, USA p 1529-1536
ISSN: 0065-2814 ISBN: 1-55617-052-1
Abstract: Simulation experiments and designs are now coming to fruition as physical systems. However, as more and more new AGV systems are modeled and then implemented, it is becoming evident that physical systems do not perform as represented in the simulation model. The problem we have found as we have compared the simulation models with the physical systems, is that the configurations and control algorithms are oftentimes misrepresented in the original design specification which was used as the basis for the model.
Descriptors: *Industrial Trucks--*Control Systems; Control Systems-- Computer Simulation
Identifiers: AGV System; Simulation Modeling; Physical Systems

Title: Factors Influencing Simulation Accuracy of AGV Networks.
Author: Quinn, Edward B.
Corporate Source: AutoSimulations Inc, Bountiful, UT, USA
Conference Title: Proceedings - 1984 National Material Handling Forum: Advanced Technologies Exposition.
Conference Location: Houston, TX, USA Conference Date: 1984 Mar 27-29
Source: Publ by Material Handling Inst Inc, Pittsburgh, PA, USA 12p
Abstract: Improved simulation techniques are available and established that permit accurate forecasting of AGV system performance. The techniques must be applied in orderly fashion and in parallel with the design definition of the system. A simulation is most beneficial if it is accomplished before a commitment is made to a design. Simulation models must be validated to establish that they accurately represent the physical system being imitated. Through use of simulations the designer can be lead to better layouts and improved AGV network performance.
Descriptors: *MATERIALS HANDLING--*Mechanization; MACHINE SHOPS-- Automation
Identifiers: Automatic Guided Vehicle (AGV); Dark Camera Film Storage Facility; Carrier Reduction; Modeling Capabilities; Computer Processing Time; Train Collision Avoidance

Title: Simulation based system for automatic development and testing of AGV control software.
Author: Quinn, E. B.
Corporate Source: AutoSimulations Inc, USA
Conference Title: Proceedings of the 3rd International Conference on Automated Guided Vehicle Systems.
Conference Location: Stockholm, Swed Conference Date: 1985 Oct 15-17
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 219-227
ISBN: 0-903608-98-7

Abstract: A simulation based software system has been developed that permits automatic development and testing of AGV control software. The simulation model receives network guidepath definition from standard Computer Aided Design systems. The model permits design testing and optimization using a literal (English) language. Outputs from the model are used to interface to an emulator that imitates network protocol to the system controller for testing of controller software. The same model can be used for direct compilation of controller software. This final feature automates the entire software development procedure. (Author abstract)
Descriptors: *Vehicles--*Automation; Computer Simulation; Computer Software
Identifiers: Automated Guided Vehicles Systems (AGVS); AGV Control Software; Software Development Procedure; Simulation Based System; Simulation Based Software System; AGV Model Functional Requirements

R

Title: Addressing design and control issues of AGV-based FMSs with Petri net aided simulation
Author: Raju, K. Ravi; Chetty, O.V. Krishnaiah
Corporate Source: Indian Inst of Technology, Madras, India
Source: Computer Integrated Manufacturing Systems v 6 n 2 May 1993. p 125-134
ISSN: 0951-5240
Abstract: Automated Guided Vehicle Systems (AGVS) are widely used in Flexible Manufacturing Systems (FMS) as they provide modularity and flexibility. These systems are highly complex and expensive, hence careful design and operational planning are essential. Realizing the limitations of the available analytical and simulation methods, this paper proposes extensions to the Timed Petri nets method, and presents an hierarchical modelling and simulation methodology. It addresses design and control issues such as the number of vehicles required, machine scheduling, AGV dispatching, etc., of an AGV-based FMS. (Author abstract) 25 Refs.
Descriptors: *Flexible manufacturing systems; Control; Automation; Petri nets; Systems analysis; Parameter estimation; Hierarchical systems; Mathematical models
Identifiers: Automated guided vehicle systems (AGVS)

Title: Design and evaluation of automated guided vehicle systems for flexible manufacturing systems: An extended timed Petri net-based approach
Author(s): Raju, K. Ravi; Chetty, O.V. Krishnaiah
Inst of Technology, Madras, India
INT J PROD RES VOL. 31, NO. 5 pp. 1069-1096 1993
Abstract: The design and evaluation of automated guided vehicle systems (AGVSs) for flexible manufacturing systems (FMSs) is complex because of the randomness and number of variables involved. A Petri net-based methodology is proposed in this paper for modelling and simulating AGVSs for FMSs. To this end, the capabilities of time Petri nets are expanded. The proposed methodology is elucidated with an example. Several design and operational and control issues, including the number of vehicles required, buffer sizes, siding sizes and vehicle dispatching, are addressed. The effect of machine and AGV scheduling rules on system performance is also investigated.

Descriptors: Materials handling equipment; Electronic guidance systems; Vehicles; Scheduling; Machine design; Control systems; Mathematical models; Petri nets; Random processes; Computer software

Title: Multi-criteria operational control rules in flexible manufacturing systems (FMSs)
Author(s): Ro, I.-K.; Kim, J.-I.
CAD/CAM Lab., Dep. Ind. Eng., Hanyang Univ., Seoul 133-791, Rep. Korea
INT. J. PROD. RES. VOL. 28, NO. 1 pp. 47-63 1990
Abstract: This paper is concerned with the operational control problems in Flexible Manufacturing Systems (FMS) with limited local buffers. Three new process selection rules which can be applied to the FMS with limited local buffers and can flexibly cope with the change of system configuration (machine breakdown, etc.) are developed. In addition, AGV dispatching and AGC route selection rules are also developed for the simultaneous scheduling of jobs and material handling devices.
Descriptors: flexible manufacturing systems; production scheduling; simulation

S

Title: Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system.
Author: Sabuncuoglu, Ihsan; Hommertzheim, Don, L.
Corporate Source: Bilkent Univ, Ankara, Turk
Source: International Journal of Production Research v 30 n 5 May 1992 p 1059-1079
ISSN: 0020-7543
Abstract: In this paper, an on-line dispatching algorithm is proposed for the FMS scheduling problem. The algorithm uses various priority schemes and relevant information concerning the load of the system and the status of jobs in the scheduling process. This information is organized into hierarchical levels. The scheduling decision process is hierarchical in the sense that different decision criteria are applied sequentially to identify the most appropriate part and the machine to be served. The algorithm schedules the jobs on a machine or an automated guided vehicle (AGV) one at a time as the scheduling decision is needed (or as the status of the system changes). Performance of the proposed algorithm is compared with several machine and AGV scheduling rules by using the mean flow-time and the mean tardiness criteria. Simulation results indicate that the proposed algorithm produces significant mean flow-time and mean tardiness improvements over existing scheduling rules for a variety of experimental conditions. (Author abstract) 32 Refs.
Descriptors: *Production Control--*Scheduling; Industrial Plants-- Flexible Manufacturing Systems; Computer Aided Manufacturing; Mathematical Techniques--Algorithms; Vehicles--Automatic Operation
Identifiers: Flexible Manufacturing System; Dynamic Dispatching Algorithm ; Automated Guided Vehicles (AGV); Mean Flow Time; Mean Tardiness

Title: Experimental investigation of FMS machine and AGV scheduling rules against the mean flow-time criterion.
Author: Sabuncuoglu, Ihsan; Hommertzheim, Don, L.
Corporate Source: Bilkent Univ, Ankara, Turk
Source: International Journal of Production Research v 30 n 7 Jul 1992 p 1617-1635
ISSN: 0020-7543
Abstract: Although a significant amount of research has been carried out in the scheduling of flexible manufacturing systems (FMSs), it has generally been focused on developing intelligent scheduling systems. Most of these systems use simple scheduling rules as a part of their decision process. While these scheduling rules have been investigated extensively for a job shop environment, there is little guidance in the literature as to their performance in an FMS environment. This paper attempts to investigate the performances of machine and AGV scheduling rules against the mean flow-time criterion. The scheduling rules are tested under a variety of experimental conditions by using an FMS simulation model. (Author abstract) 34 Refs.
Descriptors: *Flexible Manufacturing Systems--*Scheduling; Scheduling-- Optimization; Flexible Manufacturing Systems--Performance; Industrial Plants--Machinery; Computer Simulation--Industrial Applications Identifiers: Mean Flow Time Criterion

Title: Modeling AGV Systems Using Network Constructs.
Author: Sale, Michael W.; Stein, Catherine W.
Corporate Source: Pritsker & Associates Inc, West Lafayette, IN, USA
Conference Title: 1987 Winter Simulation Conference Proceedings.
Conference Location: Atlanta, GA, USA Conference Date: 1987 Dec 14-16
Source: Winter Simulation Conference Proceedings 1987. Publ by ACM, New York, NY, USA. Available from IEEE Service Cent (Cat n 87CH2512-2), Piscataway, NJ, USA p 661-668
ISSN: 0275-0708 ISBN: 0-911801-32-4
Abstract: The modeling of an automated guided-vehicle system (AGVS) used in manufacturing can be difficult because of the special nature of its resources and activities. A description is given of the Material Handling Extension for SLAM II, MHEX, which overcomes these difficulties through a specialized set of network capabilities. The unique requirements imposed by an AGVS model are described. An example which emphasizes the use of the MHEX and how it is integrated with SLAM II is provided. 4 refs.
Descriptors: *Computer Aided Manufacturing--*Equipment; Vehicles-- Computer Simulation
Identifiers: Automated Guided-Vehicle System; Material Handling Extension ; Mhex; SLAM II

Title: Material handling in a flexible manufacturing system processing part families.
Author: Savory, Paul A.; Mackulak, Gerald T.; Cochran, Jeffery K.
Conference Title: 1991 Winter Simulation Conference Proceedings
Conference Location: Phoenix, AZ, USA Conference Date: 1991 Dec 8-11
Source: Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 91CH3050-2). p 375-381
Publication Year: 1991
ISSN: 0275-0708 ISBN: 0-7803-0181-x

Abstract: The system performance of a six-machine flexible manufacturing system cell in which a material handling system is to be incorporated is analyzed. The analysis focuses on determining the production potential of the cell by grouping common parts into families. To accomplish this, computer simulation models are developed using the SIMAN simulation language. Initially the manufacturing cell is modeled with no material handling system to get an upper bound estimate of production output. The authors explore the impact that an automatic guided vehicle (AGV) has on system performance of the manufacturing system cell. A final analysis is performed in which a conveyor system is implemented as the material handling device. Upon examination of the simulation results, the authors's recommendation is to implement a conveyor system for material handling. Using an AGV in the flexible manufacturing cell creates a bottleneck which causes production output to dramatically decrease. 14 Refs.

Descriptors: Computer Simulation; Materials Handling; Flexible Manufacturing Systems; Computer Simulation Languages

Identifiers: Siman (Simulation Language); Automatic Guided Vehicle (AGV)

Title: Material handling comparisons of AGV units

Author(s): Schiller, T.E.

Eaton-Kenway, Inc.

1990 IEEE Technical Conference on Rubber and Plastics Industries 9020563 Akron, OH (USA) 30 Apr - 1 May 1990

IEEE Industry Applications Society

Bob Fromm, R.K. Fromm, Inc., P.O. Box 4224, Akron, OH 44321, USA

Title: Knowledge based systems for scheduling and control of an automated manufacturing cell

Author(s): Sepulveda, J.M.; Sullivan, W.G.

Ind. Eng. Dep., 153 Alumni Gym, Univ. Tennessee, Knoxville, TN 37996-1506, USA

COMP. IND. ENG. VOL. 15 pp. 59-66 1988

10. Annual Conference for Computers and Industrial Engineering 23-25 Mar 1988 Dallas, TX.

Abstract: Cell Manager (Cm) is a prototype knowledge based system for real-time control and short-term planning of an automated manufacturing facility. It uses background of mathematical programming, simulation, and heuristic search. The Cm engine allows tracking of time related events and facts in discrete or continuous time. The use of Cm is illustrated on a cell that processes discrete batches of parts. There are four workstations and the material handling system consists of automated guided vehicles.

Descriptors: production scheduling; computer aided control; computer integrated manufacturing; simulation

Title: An analytical method for configuring fixed-path, closed-loop material handling systems
Author(s): Sharp, G.P.; Liu, F-H.F.
Mater. Handl. Res.Cent., Sch. Ind. & Syst. Eng., Georgia Inst. Technol., Atlanta, GA 30332, USA
INT. J. PROD. RES. VOL. 28, NO. 4 pp. 757-783 1990
Abstract: This paper presents an analytical method for configuring the network of a fixed-path, closed-loop material handling system. The method is applicable to towlines, automated monorail systems (AMS), automated guided vehicles systems (AGVS), and other equipment types. The purpose is to make good initial decisions with respect to adding shortcuts (cutbacks), adding off-line spurs, and their length, so that of the thousands or millions of possible network configurations only a very small number need subsequently to be examined by simulation. The approach taken is to develop volume-delay curves for load-carriers passing a load/unload station, based on average transport requirements. Future work in this area is planned with two objectives in mind: simplifying and speeding up the analytical method presented in this report; and linking the analytical method with a simulation procedure.
Descriptors: mechanical handling; automated guided vehicles; production design; production costs

Title: Intelligent Factory Transport System.
Author: Solberg, J. J.; McGillem, C. D.
Corporate Source: Purdue Univ, USA
Conference Title: Proceedings of the 3rd International Conference on Automated Materials Handling.
Conference Location: Birmingham, Engl Conference Date: 1986 Mar 19-21
Source: Publ by IFS (Publications) Ltd, Kempston, Engl p 343-350
ISBN: 0-948507-10-1
Abstract: The goals of the system are to provide a high degree of flexibility while at the same time permitting a major reduction of work-in-process inventory. The system employs a fleet of small, fast robot vehicles able to travel freely throughout the factory. The vehicles would be in constant communication with a central computer that would dispatch them on demand. The vehicles would handle individual work pieces or in some cases two or three units at a time. Individual vehicles would have sufficient on-board intelligence to navigate throughout the work space to detect obstacles, avoid collisions, maintain speed and so forth. The central computer would optimize utilization of the vehicles and the machines in a flexible manufacturing system. The various tasks involved in a research project designed to prove the feasibility of this system are described. 17 refs.
Descriptors: *Industrial Trucks--*Automation; Industrial Plants--Flexible Manufacturing Systems; Materials Handling--Computer Simulation; Robots, Industrial--Intelligent
Identifiers: Intelligent Factory Transport System; Work-In-Process Inventory; Automated Guided Vehicles; FMS

T

Title: Module modeling and economic optimization for large-scale AS/RS.
Author: Takakuwa, Soemon
Corporate Source: Sch of Bus Admin, Toyo Univ, Bunkyo-ku, Tokyo, Japan
Conference Title: 1989 Winter Simulation Conference Proceedings - WSC '89
Conference Location: Washington, DC, USA Conference Date: 1989 Dec 4-6
Source: Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 89CH2778-9). p 795-801
ISSN: 0275-0708
Abstract: A method of decomposing models for large-scale automated storage/retrieval systems (AS/RSs) is proposed in an attempt to determine the optimum combination of warehouses and material handling systems. The AS/RS considered comprises the automated warehouse with stacker cranes, the conveyance system, and the handling systems. There are three alternatives in the conveyance system, i.e., single-track AGV (automated guided vehicle) systems, looped-track AGV systems, and conveyor systems. The effectiveness of the approach is confirmed by performing both simulation and animation of models based upon an actual example. A procedure for obtaining the optimum combination of subsystems for constructing AS/RS warehouses is presented, with numerical examples, from an economic standpoint. 5 Refs.
Descriptors: Computer Simulation-Applications; Warehouses; Cranes; Optimization; Economics
Identifiers: Automated Storage/Retrieval Systems; AGV (Automated Guided Vehicle) Systems

Title: Study on decision rules of a scheduling model in an FMS
Author: Tang, Ling-Lang; Yih, Yuehwern; Liu, Chang-Yung
Corporate Source: Natl Sun Yat-Sen Univ, Kaohsiung, Taiwan
Source: Computers in Industry v 22 n 1 Jun 1993. p 1-13
ISSN: 0166-3615
Abstract: Flexible manufacturing systems (FMs) are designed to combine the efficieny of a mass-production line and the flexibility of a job shop for the batch production of a mid-volume and mid-variety of products. To control FMSs is more complex than transfer lines or job shops because of the flexibility of machines and operations. In general, FMS operation decisions can be divided into two phases: planning and scheduling. The planning phase considers the pre-arrangement of parts and tools before the FMS begins to process, and the scheduling phase deals with routing parts while the system is in operation. This paper focuses on the scheduling phase. We establish six decision points involved in the operation among parts, AGVs and machines. The dispatching rules associated are evaluated based on simulation experiments. The performance measurement is defined as a weighted index of multicriteria, including time in system, mean tardiness, maximum tardiness, system utilization, number of machines locked, buffer size, and work in process. The Taguchi method is used in the simulation experiments to study the relationship between the multiple performance measures and the dispatching rules in six decisions points (Author abstract) 21 Refs.
Descriptors: *Flexible manufacturing systems; Supports; Computer simulation; Vehicles
Identifiers: Multi-criteria; Automated guided vehicles

Title: Modeling AGV systems using network constructs
Author(s): Thesen, A.; Grant, H.; Kelton, W.D. (eds.); Sale, M.W.; Stein, C.W.
Pritsker and Assoc., Inc., 1305 Cumberland Ave., West Lafayette, IN 47906, USA
1987 WINTER SIMULATION CONFERENCE. pp. 661-668 1987
1987 Winter Simulation Conference 14-16 Dec 1987 Atlanta, GA (USA)
Publ: Society For Computer Simulation, P.O. BOX 17900, SAN DIEGO, CA 92117-7900 (USA),
Abstract: The network world view of SLAM II) has been demonstrated as an effective method for modeling manufacturing processes. An automatic guided vehicle system (AGVS), however, may be difficult because of the special nature of its resources and their activities. This paper presents the Material Handling Extension for SLAM II nu (MHEX). The MHEX overcomes these difficulties through a specialized set of network capabilities. This paper begins by describing the unique requirements imposed by an AGVS model. Next, the MHEX modeling approach is presented relative to these requirements. The paper is concluded by an example which emphasized the use of the MHEX and how it is integrated with SLAM II.
Descriptors: manufacturing; modeling; control systems; mechanical handling; guidance and control systems

Title: Flexible Manufacturing For Limited Production Of Concentrics.
Author: Thettu, R. R.; Paidy, S. R.
Corporate Source: Xerox Corp
Conference Title: Proceedings of the 1st International Conference on Simulation in Manufacturing.
Conference Location: Stratford-upon-Avon, Engl Conference Date: 1985 Mar 5-7
Source: Publ by IFS (Publ) Ltd, Kempston, Engl p 239-250
ISBN: 0-903608-84-7
Abstract: To gain an indepth understanding of the intricacies of a flexible manufacturing system for the production of concentric parts in Xerox copiers and to specify adequate equipment, human, computer, and material handling resources before implementation, a simulation model has been developed. The newly emerged SIMAN language is used to describe the complex manufacturing environment comprising nine work centers and automated guided vehicle (AGVS) material handling system. A process-oriented modeling approach along with Fortran extensions for the control logic of the three machining centers have provided a modular and easy to understand simulation model to analyze several design options and estimate overall production capacity and equipment utilization. Future uses of such model include realtime/dynamic simulation to predict the product delivery times and impacts of priority changes in production schedules without burdening the actual FMS control system. (Edited author abstract) 3 refs.
Descriptors: *Systems Engineering--*Computer Simulation; Industrial Plants--Flexible Manufacturing Systems; Production Control--Scheduling; Production Engineering--Design Aids; Systems Science And Cybernetics
Identifiers: Fms; Concentric Parts Production; System Design

Title: Simulation of Integrated Material Handling Systems in an FMS Environment - A Case History.
Author: Tracy, Michael J.
Corporate Source: Smith, Hinchman & Grylls Associates Inc, Detroit, MI, USA
Conference Title: AUTOFACT '86.
Conference Location: Detroit, MI, USA Conference Date: 1986 Nov 12-14
Source: Publ by SME, Dearborn, MI, USA p 2. 61-2. 74
ISBN: 0-87263-258-X
Abstract: In planning of automated factories, many decisions regarding the complex interaction between systems is made. The use of simulation as a tool in this decision process is essential to development of optimized systems that maintain flexibility and adaptability for change in the dynamic world of manufacturing. This paper presents a case history of a simulation project for a flexible manufacturing system (FMS) facility in the aerospace industry. The principal feature is the integration of an automated storage and retrieval system (AR/RS) and an automatic guided vehicle system (AGVS) into the FMS environment using simulation for: physical layout evaluation for capacity, analysis of system expandability, and evaluation of alternative scheduling strategies. (Author abstract)
Descriptors: *Materials Handling--*Computer Simulation; Vehicles-- Automation; Aircraft Materials--Storage; Aerospace Engineering--Computer Applications; Industrial Plants--Flexible Manufacturing Systems
Identifiers: Flexible Manufacturing System (FMS); Automatic Guided Vehicle System (AGVS); Material Move Logic

U

Title: Using simulation in design of a cellular assembly plant with automatic guided vehicles.
Author: Ulgen, Onur M.; Kedia, Pankaj
Corporate Source: Production Modeling Corp, Dearborn, MI, USA
Conference Title: 1990 Winter Simulation Conference Proceedings
Conference Location: New Orleans, LA, USA Conference Date: 1990 Dec 9-12
Source: 90 Winter Simulation Conf. Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 90CH2926-4). p 683-691
ISSN: 0275-0708 ISBN: 0-911801-72-3
Abstract: The effects of a large number of design and operational variables on the performance of a relatively complex cellular assembly system with automatic guided vehicles (AGVs) are analyzed. The techniques used by previous researchers in design of automatic guided vehicle systems (AGVS) are classified into three categories: optimization techniques, mathematical heuristic techniques, and simulation techniques. Factors to be considered in an AGVS are classified into eight levels hierarchically, the highest level being the process focus and the lowest level being the schedule-related considerations. The cellular assembly system considered is a simplified version of a real AGVS previously investigated. The operational variables studied include AGV and job dispatching rules and assembly time ratios between the mainline and subassembly processes. The study showed that effective use of transit paths in a track layout design may eliminate the differences among the scheduling rules. The most significant scheduling rule in increasing the throughput is the job request selection rule by an AGV for a design with a minimum number of transit paths. 20 Refs.
Descriptors: *Computer Simulation--*Industrial Applications; Computer Aided Manufacturing
Identifiers: Automated Guided Vehicle Systems(AGVS); Cellular Assembly Systems

Title: AGV flow path design and load transfer point location
Author(s): Usher, J.S.; Evans, G.W.; Wilhelm, M.R.
Dep. Ind. Eng., Univ. Louisville, Louisville, KY
International Industrial Engineering Conference 8825007 Orlando, FL (USA) 22-25 May 1988
Institute of Industrial Engineers, 25 Technology Park/Atlanta, Norcross, GA 30092 (USA), ISSN 0887-4719; Order No.IIE-P-293
Descriptors: Civil And Mechanical Engineering

V

Title: Simulation study of an AGV system in an FMS environment
Author(s): Vosniakos, G.-C.; Davies, B.J.
Div. Manuf. and Mach. Tools, Dep. Mech. Eng., UMIST, P.O. Box 88, Manchester M60 1QD, UK
INT. J. ADV. MANUF. TECHNOL. VOL. 3, NO. 4 pp. 33-46 1988
Abstract: A simulation program written in ECSL to simulate an AGV system serving an FMS is presented. It incorporates a special algorithm to enable "intelligent" routing of the AGVs with minimum control requirements. The program was used to examine several parameters and scheduling/control disciplines affecting the AGV system's performance for a specific FMS. Results show that selection of an appropriate combination of parameter values can have a beneficial effect on the FMS.
Descriptors: automated guided vehicles; flexible manufacturing systems; routing; simulation

Title: Automated guided vehicle system design for FMS applications
Author(s): Vosniakos, G.-C.; Mamalis, A.G.
Dep. Mech. Eng., MMT Div., UMIST, Manchester M60 1QS, UK
INT. J. MACH. TOOLS MANUF. VOL. 30, NO. 1 pp. 85-97 1990
Abstract: The issues of designing and installing a system of Automated Guided Vehicles (AGVs) in a Flexible Manufacturing System (FMS) are examined. The development, advantages and future trends of AGVs are briefly reviewed. Then the basic features of an AGV in an FMS environment, as well as computerized procedure for the optimum vehicle selection, are discussed. Guide path layout considerations are made and docking/pallet transfer methods are suggested. A routing algorithm is presented to facilitate both routing and control of the vehicles, and most commonly used methods for zone blocking, loading/unloading and traffic control and communications are outlined. Operation rules are reviewed and an example of an AGVS in an FMS is presented to demonstrate the potential of simulation in both design and evaluation of such systems.
Descriptors: flexible manufacturing systems; advanced manufacturing technology; automated guided vehicles; design; control systems

W

Title: Animated Graphic Simulation of an Automatic Guided Vehicle System (AGVS).
Author: Wang, Shay-Ping T.
Corporate Source: Intel Corp, Chandler, AZ, USA
Conference Title: 1985 Winter Simulation Conference Proceedings.
Conference Location: San Francisco, CA, USA Conference Date: 1985 Dec 11-13
Source: Winter Simulation Conference Proceedings 1985. Publ by IEEE, New York, NY, USA. Available from IEEE Service Cent (Cat n 85CH2214-5), Piscataway, NJ, USA p 252-256
ISSN: 0275-0708 ISBN: 0-911801-07-3
Abstract: The graphic simulation package PCMODEL is successfully applied to the planning of an automatic guided vehicle system (AGVS) in one of Intel's facilities. The AGVS is controlled by one central computer and consists of a number of docks from which materials are either dropped or picked up by the AGV; the dispatch of AGV is prioritized to efficiently perform material handling. The utilization of AGV can be obtained through simulation, and is used to predict the number of AGV in the system. The AGV can carry two loads, therefore, it is poorly utilized if it only transports one load a trip. To better utilize AGV, the control delay concept is introduced, which forces the AGV to carry as many two loads as possible at each trip. The simulation also provides guideline on the selection of control delay. 9 refs.
Descriptors: *Computer Simulation; Vehicles; Computer Aided Manufacturing ; Computer Graphics
Identifiers: Animation; Manufacturing Automation; PC Model Graphic Simulation Package; Automatic Guided Vehicle System

Title: Animated Simulation of a Flexible Manufacturing System.
Author: Wang, Shay-Ping
Corporate Source: Intel Corp, Chandler, AZ, USA
Conference Title: 1986 Winter Simulation Conference Proceedings.
Conference Location: Washington, DC, USA Conference Date: 1986 Dec 8-10
Source: Winter Simulation Conference Proceedings 1986. Publ by IEEE, New York, NY, USA. Available from IEEE Service Cent (Cat n 86CH2385-3), Piscataway, NJ, USA p 633-640
ISSN: 0275-0708 ISBN: 0-911801-11-1
Abstract: A flexible manufacturing system (FMS) requires computer control of a variety of machines, complicated product routings and mixing, and costly installation. To reduce potential design errors and associated financial risk, simulation techniques must be utilized. The author applies animated simulation to a flexible robotic inspection cell, a flexible wave soldering cell, and an automated guided vehicle system (AGVS) in an integrated circuit assembly plant in Arizona. The flexible robotic inspection cell consists of two inspection stations, two robots, a programmable conveyor system, and input/output mechanisms. The simulation identifies the units per hour of the cell, and operator's utilization which gives clues to multiple cell handling. The flexible wave soldering cell includes a wave soldering machine, a loader, and an unloader. The simulation provides insights to critical design parameters of the loader/unloader and the number of operators that are necessary to maintain smooth operation of the cell. In the AGVS, one automated guided vehicle (AGV) and 21 docks are used. The AGV picks up or drops off a tote of materials on each dock. Excessive raw material delivered to each dock increases manufacturing cost; however, too little will increase the number of AGV. The simulation is used to identify the optimal quantity of material in a tote. 11 refs.

Descriptors: *computer aided manufacturing--*computer simulation; inspection--automation; vehicles--automation; robots, industrial; soldering --automation; materials handling
Identifiers: Automated Guided Vehicles; Animated Simulation; Flexible Robotic Inspection Cell

Title: Building flexible AGV and ASRS system models for facility design phase applications.
Author: Webster, Ronald L.; Foster, David F.
Corporate Source: Inland Steel Flat Products Co, East Chicago, IN, USA
Conference Title: 1990 Winter Simulation Conference Proceedings
Conference Location: New Orleans, LA, USA Conference Date: 1990 Dec 9-12
Source: 90 Winter Simulation Conf. Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 90CH2926-4). p 692-698
Publication Year: 1990
ISSN: 0275-0708 ISBN: 0-911801-72-3
Abstract: The authors review important input, output, and model integration factors to consider when developing detailed flexible models of AGV (automatic guided vehicle) and ASRS systems. Examples are presented for setting up a model using the SIMAN simulation language, with Fortran enhancements. The design and development of an AGV and ASRS simulation that provides for the input and output characteristics outlined result in a model system that is both flexible and comprehensive. AGVs and ASRS cranes can be added or removed. AGV and ASRS path segments and routes can be easily modified or additional ones can be incorporated. Loads can be assigned to ASRS bays by load type, and inventory initialization levels can be easily revised. Material transporter equipment specifications can be readily modified. Capturing the output described, in addition to pertinent facility production data, allows thorough system performance analysis. If other parameters are found to be significant, the statistical and trend output generated is easily expandable. 2 Refs.
Descriptors: *Computer Simulation--*Industrial Applications; Computer Aided Manufacturing; Computer Simulation Languages--Applications Identifiers: Simulation Language Siman; Automated Guided Vehicle Systems(AGVS)

Title: A survey of varied production systems and different aspects using computer simulation.
Author: Weigl, K. Heinz
Corporate Source: Inst of Manuf Eng, Tech Univ Vienna, Austria
Conference Title: 1991 Winter Simulation Conference Proceedings
Conference Location: Phoenix, AZ, USA Conference Date: 1991 Dec 8-11
Source: Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA (IEEE cat n 91CH3050-2). p 465-473
ISSN: 0275-0708 ISBN: 0-7803-0181-x
Abstract: The author presents different problems associated with conducting simulation projects of a wide range of production systems. He provides some insight into the variety of initial conditions and illustrates different situations for the modeler. It is pointed out that historically grown systems, e.g., in the metal industry with a very low level of automation and poorly controlled material flow, are very difficult to analyze and to develop the functional specification for unpredictable benefits for the production engineer during the simulation process. AGV (automatic guided vehicle) modeling in the automotive industry and a flexible manufacturing system are among the examples considered. All finished projects described used SIMAN/CINEMA for model development. 7 Refs.
Descriptors: *Computer Simulation; Production Engineering; Industrial Applications
Identifiers: Simulation Modeling; Production Systems; Simulation Model Development

Title: Modeling AGV systems
Author(s): Wilson, J.R.; Henriksen, J.O.; Roberts, S.D. (eds.); Davis, D.A.
Syst. Model. Corp., Calder Sq., P.O. Box 10074, State College, PA 16805, USA
WSC '86. pp. 568-574 1986
1986 Winter Simulation Conference 8-10 Dec 1986 Washington, DC (USA)
Publ: Association For Computing Machinery, 11 West 42nd Street, New York, NY 10036 (Usa),
Computer simulation is often used as an analysis tool during the design of Automated Guided Vehicle (AGV) systems. However, because of the complexities inherent in automated material handling systems, general-purpose simulation languages must be used creatively to capture the desired detail in the model. This paper presents some general concepts which can be used to model AGV systems. Also, some of the critical concerns which must be addressed in a simulation analysis of an AGV system are presented and discussed.
Descriptors: manufacturing; mechanical handling; automation; vehicles; simulation; modeling

Title: Animated simulation of a flexible manufacturing system
Author(s): Wilson, J.R.; Henriksen, J.O.; Roberts, S.D. (eds.); Wang, S.P.T.
Intel Corp., 145 South 79th St., Chandler, AZ 85226, USA
WSC '86. pp. 633-640 1986
1986 Winter Simulation Conference 8-10 Dec 1986 Washington, DC (USA)
Publ: ASSOCIATION FOR COMPUTING MACHINERY, 11 WEST 42ND STREET, NEW YORK, NY 10036 (USA),
Abstract: A flexible manufacturing system requires computer control of a variety of machines, complicated product routings and mixing, and costly installation. To reduce potential design errors and associated financial risk, simulation techniques must be employed. This paper applies animated simulation to a flexible robotic inspection cell, a flexible wave soldering cell, and an automated guided vehicle system (AGVS) in one of Intel's integrated circuit assembly plants in Arizona.
Descriptors: flexible manufacturing systems; simulation; computer aided manufacturing; robots; assembling; optimization; electronic equipment

Y

Title: Push and pull rules for dispatching automated guided vehicles in a flexible manufacturing system
Author(s): Yim, Dong-Soon; Linn, R.J.
Iowa State Univ, Ames, IA, USA
INT J PROD RES VOL. 31, NO. 1 pp. 43-57 1993
Abstract: Automated guided vehicle (AGV) systems are widely used in flexible manufacturing systems (FMSs) for material handling purposes. Although the AGV systems have provided high flexibility, the design issues on AGV dispatching rules are still to be resolved. The AGV-dispatching rules in an FMS are generally based either on a push or a pull concept. A Petri-net-based simulation was used to investigate the effect of different dispatching rules on the FMS performance. The developed simulation model consists of two modules: a Petri net model and an AGV dispatcher. The two modules are integrated so that the AGV dispatcher controls AGV tokens in the Petri net model. It was shown that there was no significant difference in output rate between push- and pull-based AGV-dispatching rules for a busy FMS.
Descriptors: Factory automation; Guideways; Vehicles; Flexible manufacturing systems; Scheduling; Optimization; Hierarchical systems; Control systems; Petri nets; Computer *simulation*

Z

Title: Conflict detection of automated guided vehicles. A Petri net approach.
Author: Zeng, Laiguang; Wang, Hsu-Pin; Jin, Song
Corporate Source: Rochester Inst of Technology, Rochester, NY, USA
Source: International Journal of Production Research v 29 n 5 May 1991 p 865-879
ISSN: 0020-7543
Abstract: This research project investigates and develops techniques for automatically detecting potential vehicle conflicts in an automated guided vehicle (AGV) system. Three activities are involved: (1) construction of a set of formal definitions of Petri nets for modelling AGV systems; (2) development of procedures for detection of vehicle routing conflicts; (3) development of a simulation program of the procedure. In this paper the first two activities are discussed in detail, and examples are provided. (Author abstract) 42 Refs.
Descriptors: *Materials Handling--*Automation; Vehicles--Guideways; Transportation--Route Analysis; Computer Simulation
Identifiers: Automated Guided Vehicles; Petri Nets; Vehicle Routing Conflicts

Appendix B:

Extended Bibliography for Intelligent Simulation Modeling

A

[1]
Title: The array processor as an intelligent simulation co-processor.
Author: Alexander, Peter
Corporate Source: CPS Inc, Burlington, Mass
Source: Proc Summer Comput Simul Conf 1979 SCSC, Toronto, Ont, Jul 16-18 1979. Available from AFIPS Press, Montvale, NJ p 963-967
Publication Year: 1979
ISSN: 0094-7474
Abstract: A description is given of the use of array processors as high speed intelligent peripheral devices for a minicomputer based simulation system. The Macro Arithmetic Processor (MAP) is presented as an example, and a review of the hardware and software architectural features is given. The internal multiprocessor structure of MAP is highlighted, and the way in which these programmable units interact is discussed. An overview of simulation applications is given.

[2]
Title: Another approach to AI: intelligent simulation environments
Author(s): Alvarado, F.L.; Liu, Y.
Author Affiliation: Dept. of Electr. & Comput. Eng., Wisconsin Univ., Madison, WI, USA
Conference Title: Fifth International Conference on Systems Engineering (Cat. #.87CH2480-2) p.295-8
Publisher: IEEE, New York, NY, USA
Publication Date: 1987 Country of Publication: USA xiii+593 pp.
U.S. Copyright Clearance Center Code: CH2480-2/87/0000-0295$01.00
Conference Date: 9-11 Sept. 1987 Conference Location: Dayton, OH, USA
Abstract: The application of artificial intelligence (AI) to simulation in electric power systems, with extensions to engineering simulation in general, is explored. Previous research efforts have been concerned with the use of expert systems in solving knowledge-intensive, nonalgorithmic power systems problems. An alternative approach based on intelligent simulation environments is presented, in which the objective is to use AI tools to assist in the solution of well-defined but complex tasks. A prototype intelligent simulation environment is introduced and is used to solve a variety of electric power systems simulation problems. (10 Refs)
Descriptors: artificial intelligence; digital simulation; power system analysis computing
Identifiers: electric power system simulation; digital simulation; power system analysis computing; AI; intelligent simulation environments; artificial intelligence

B

[3]
Title: Coupling symbolic and numerical computation for intelligent simulation
Author(s): Beale, G.O.; Kawamura, K.
Author Affiliation: Dept. of Electr. & Comput. Eng., George Mason Univ., Fairfax, VA, USA
Book Title: Knowledge-based system diagnosis, supervision and control p.137-52
Editor(s): Tzafestas, S.G.
Publisher: Plenum, New York, NY, USA
Publication Date: 1989 Country of Publication: USA xii+305 pp.
ISBN: 0 306 43036 3
Abstract: Expert systems technology has been gaining in popularity and acceptance in the engineering community. Due to the need for results generation and results interpretation in complex problems, there is a need to couple symbolic and numerical techniques. Such coupled systems promise to integrate the explanation and problem solving capabilities of knowledge-based systems with the precision of traditional numerical computing. One application area where this coupling of symbolic and numerical computing will have great benefit is the digital simulation of complex dynamic systems. The authors describe the design principle and architecture of one such coupled system, named NESS (NASA Expert Simulation System). NESS assists the user in running digital simulations of dynamic systems; interprets the output data to determine system characteristics; and, if the output does not meet the performance specifications, recommends a suitable series compensator to be added to the simulation model. (14 Refs)
Descriptors: digital simulation; engineering computing; expert systems; explanation; problem solving
Identifiers: symbolic computation; integration; numerical computation; intelligent simulation; engineering; explanation; problem solving; knowledge-based systems; precision; digital simulation; complex dynamic systems; NESS; NASA Expert Simulation System

[4]
Title: AUDITION-an intelligent simulation environment
Author(s): Begg, I.M.; Worsley, C.P.
Author Affiliation: Synaptec Software, Vancouver, BC, Canada
Conference Title: Simulation and AI, 1989. Proceedings of the SCS Western Multiconference p.71-6
Editor(s): Webster, W.
Publisher: SCS, San Diego, CA, USA
Publication Date: 1989 Country of Publication: USA vii+138 pp.
ISBN: 0 911801 44 8
Conference Date: 4-6 Jan. 1989 Conference Location: San Diego, CA, USA
Abstract: Discuss the importance of closely integrating knowledge representation with the simulation environment in the production of user-customizable simulations. A specific simulation example is presented using AUDITION, a PC based intelligent simulation and visual programming environment. These simulations are rich micro-worlds incorporating both discrete and continuous evaluation. Once constructed, simulations can be reused whole, or in part, in different situations. All simulations are capable of being customized by end-users. Thus, generic components are simply reconfigured to fit into the users' environments. Knowledge representation in AUDITION is based on a Stage/Performer paradigm. The Stage/performer paradigm combines the classification networks of object-oriented representation (Performers) with the orthogonal concept of 'worlds' (Stages). A detailed example is given concerning air traffic simulation. (20 Refs)

Descriptors: air traffic; knowledge representation; object-oriented programming; programming environments; simulation languages Identifiers: PC-based simulation environment; AUDITION; intelligent simulation environment; knowledge representation; visual programming environment; micro-worlds; Stage/Performer paradigm; classification networks; object-oriented representation; air traffic simulation

[5]
Title: Intelligent simulation for flexible manufacturing systems: an integrated approach
Author(s): Benjaafar, S.
Author Affiliation: Sch. of Ind. Eng., Purdue Univ., West Lafayette, IN, USA
Journal: Computers & Industrial Engineering vol.22, no.3 Jul 1992 p.297-311
Publication Date: July 1992 Country of Publication: UK
ISSN: 0360-8352
U.S. Copyright Clearance Center Code: 0360-8352/92/$5.00+0.00
Abstract: The use of simulation in the design and control of manufacturing systems has become increasingly popular in the last few years. Although simulation has proven to be superior to traditional mathematical modeling in capturing the scope, complexity and dynamics of manufacturing systems, it remains merely a descriptive modeling tool. This leaves the user with the difficult tasks of generating, testing, and selecting design and operation alternatives. A methodology that describes the supplementation of simulation systems with these more intelligent and prescriptive functions is proposed. A new architecture, based on a structured object representation (SOR) paradigm, is suggested. It is shown that the adoption of an SOR paradigm results in easier and more effective low level integration of the modeling, simulation and reasoning functions. The design of an experimental prototype system that allows automatic model generation and simulation as well as model evaluation is also present. (31 Refs)
Descriptors: digital simulation; flexible manufacturing systems; knowledge based systems
Descriptors: *Flexible Manufacturing Systems--*Simulation; Flexible Manufacturing Systems--Design; Flexible Manufacturing Systems--Control
Identifiers: flexible manufacturing systems; integrated approach; manufacturing systems; descriptive modeling tool; simulation systems; structured object representation; reasoning functions; automatic model generation; model evaluation
Identifiers: Intelligent Simulation; Prescriptive Functions; Structured Object Representation Paradigm

[6]
Title: Using artificial intelligence techniques for intelligent simulation in memory re-education
Author(s): Bonnet, C.; Hacid, M.-S.; Taterode, H.; Kouloumdjian, J.
INSA, Dep. Inf., France
Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-90) 9030718 Charleston, SC (USA) 15-18 Jul 1990
Association for Computing Machinery - Sigart; University of South Carolina; University of Tennessee Space Institute. Dr. Moonis Ali, University of Tennessee Space Institute, M/S 15, B.H. Goethert Parkway, Tullahoma, TN 37388-8897, USA., Full Papers; ISBN: 0-89791-372-8; $72.00 plus shipping - $3.00 U.S. & Canada, $10.00 others

C

[7]
Title: Estimating a project's completion time distribution using intelligent simulation methods (PERT techniques yield highly biased estimates)
Author(s): Cook, T.M.; Jennings, R.H.
Author Affiliation: Univ. of Tulsa, Tulsa, OK, USA
Journal: Journal of the Operational Research Society vol.30, no.12 p.1103-8
Publication Date: Dec. 1979 Country of Publication: UK
ISSN: 0160-5682
Abstract: Other researchers have indicated the theoretical problems of using the PERT techniques commonly described in elementary operational research and operations management texts. Monte Carlo simulation has often been suggested as an alternative means of analyzing PERT networks but simulating a network of real world proportions is often assumed to be prohibitively expensive. The paper measures the accuracy and cost of standard PERT and simulation methods for real world sized problems. In addition, several computational heuristics are described and tested that indicate that simulation is a viable alternative. The results indicate that intelligent simulation of PERT networks is considerably more accurate than standard PERT analysis and is definitely not cost prohibitive. (15 Refs)

D

[8]
Title: Extending the mathematics in qualitative process theory
Artificial intelligence, simulation & modeling
Author(s): D'Ambrosio, B.
Contributors: Widman, Lawrence E.; Loparo, Kenneth A.; Nielsen, Norman R.
1989, 133-158 ISBN: 0-471-60599-9
John Wiley & Sons, Inc., New York, NY; Price: $44.95
Subfile: GCL (ACM Guide to Computing Literature) ACM
Descriptors: *I.6.3 -Computing Methodologies- Simulation And Modeling- Applications ; G.2.2 - Mathematics Of Computing- Discrete Mathematics- Graph Theory; H.1.1 -Information Systems- Models And Principles- Systems And Information Theory- General Systems Theory; J.2 - Computer Applications- Physical Sciences And Engineering- Engineering; I.2.8 -Computing Methodologies- Artificial Intelligence- Problem Solving, Control Methods, And Search
Identifiers: Algorithms Design Theory Verification

E

[9]
Title: Structures of discrete event simulation: an introduction to the engagement strategy.
Author(s): Evans, John B. (Univ. of Hong Kong,Hong Kong,
Publ: Halsted Press, New York, NY
1988, 279 pp. ISBN: 0-470-21097-4
Series: Ellis Horwood series in artificial intelligence.
Price: $39.95
Subfile: CR (Computing Reviews) ACM
Abstract Length: long (49 lines)
This 279-page textbook contains nine chapters: ``Models and Computation,'' ``Time Advance,'' ``Scheduling Future Events,'' ``Strategies of Discrete-Event Simulation,'' ``Simulating Chance,'' ``Simulation Language,'' ``Simulation Complexity,'' ``The SIMIAN Project,'' and ``Future Prospects of Simulation.'' It is a good text on the simulation model implementation aspects of a simulation study. The first chapter introduces some basic concepts of models and modeling, describes the development of simulation practice, and gives a short history of simulation programming. Chapter 2 describes the time flow mechanisms for discrete-event, continuous, and combined simulations. Chapter 3 nicely discusses future event scheduling algorithms and compares the methods. Chapter 4 covers event scheduling, the three-phase approach, and process interaction conceptual frameworks (also called world views or simulation strategies) and briefly compares these techniques. Unfortunately, the author fails to include the pure state-based activity scanning conceptual framework (the two-phase approach) in this chapter. Chapter 5 contains a variety of topics under the heading ``Simulating Chance.'' It briefly introduces random number and variate generation, Monte Carlo simulation, input data modeling, verification, experimentation, validation, and output analysis. Chapter 6 introduces some model representation formats and discusses the life cycle of a simulation study but does not mention any credibility assessment stage. It goes on to describe and compare some simulation programming languages. Chapter 7 introduces simulation complexity; it presents some complexity metrics and then applies them for some simulation programming languages. A subsection of Chapter 8 finally discusses the topic implied by the subtitle, introducing the author's conceptual framework for simulation model implementation, the ``engagement strategy,'' as a combination of the event scheduling, activity scanning, and process interaction frameworks. Chapter 9 briefly discusses intelligent simulation, simulation in artificial intelligence, programming paradigms, and distributed simulation. The references are rich and appropriate, and the index is satisfactory. The book lacks exercises and contains only a few examples. Evans devotes only two pages to verification and validation and does not comprehensively cover the whole life cycle of a simulation study. The book is well written, however, and could be adopted as a second textbook for a simulation course in any discipline in order to teach the simulation model implementation aspects of a simulation study.
Reviewer: O. Balci Blacksburg, VA

F

[10]
Title: Process abstraction in simulation modeling
Artificial intelligence, simulation & modeling
Author(s): Fishwick, P. A.
Contributors: Widman, Lawrence E.; Loparo, Kenneth A.; Nielsen, Norman R.
1989, 93-131
ISBN: 0-471-60599-9
John Wiley & Sons, Inc., New York, NY;
Price: $44.95
Subfile: GCL (ACM Guide to Computing Literature) ACM
Descriptors: *I.6.4 -Computing Methodologies- Simulation And Modeling- Model Validation And Analysis ; F.1.1 -Theory Of Computation- Computation By Abstract Devices- Models Of Computation; I.6.2 -Computing Methodologies - Simulation And Modeling- Simulation Languages
Identifiers: Algorithms Design Languages Verification

H

[11]
Title: Issues in qualitative reasoning about diffusional processes
Artificial intelligence, simulation & modeling
Author(s): Hardt, S. L.
Contributors: Widman, Lawrence E.; Loparo, Kenneth A.; Nielsen, Norman R.
1989, 301-323 ISBN: 0-471-60599-9
John Wiley & Sons, Inc., New York, NY;
Price: $44.95
Subfile: GCL (ACM Guide to Computing Literature) ACM
Descriptors: *I.2.8 -Computing Methodologies- Artificial Intelligence- Problem Solving, Control Methods, And Search ; I.6.3 -Computing Methodologies- Simulation And Modeling- Applications; I.2.3 -Computing Methodologies- Artificial Intelligence- Deduction And Theorem Proving- Nonmonotonic Reasoning And Belief Revision; J.2 -Computer Applications- Physical Sciences And Engineering; H.1.1 -Information Systems- Models And Principles- Systems And Information Theory- General Systems Theory
Identifiers: Algorithms Design Theory Verification

J

[12]
Title: Intelligent simulation environment for optimal control systems
Author(s): Jiang, T.-S.; Rao, M.; Tsai, J.J.P.; Hsueh, D.
Univ. Illinois, Chicago
1987 Summer Computer Simulation Conference 8735011 Montreal, Quebec (Canada) 27-30 Jul 1987

Society for Computer Simulation, P.O. Box 17900, San Diego, CA 92117. Telephone: (619) 277-3888

K

[13]
Title: Coordinating the use of qualitative and quantitative knowledge in declarative device modeling
Artificial intelligence, simulation & modeling
Author(s): Karp, P. D. , Friedland, P.
Contributors: Widman, Lawrence E.; Loparo, Kenneth A.; Nielsen, Norman R.
1989, 189-206 ISBN: 0-471-60599-9
John Wiley & Sons, Inc., New York, NY;
Price: $44.95
Subfile: GCL (ACM Guide to Computing Literature) ACM
Descriptors: *I.6.3 -Computing Methodologies- Simulation And Modeling- Applications ; J.3 - Computer Applications- Life And Medical Sciences- Biology
Identifiers: Algorithms Design Performance Theory Verification Experimentation

[14]
Title: Intelligent simulation environments for systems modeling
Author(s): Khoshnevis, B.; Austin, W.M.; Chen, A.-P.; Chen, Q.-M.
Ind. and Syst. Eng., Univ. Southern California, Los Angeles, CA
International Industrial Engineering Conference 8825007 Orlando, FL (USA) 22-25 May 1988
Institute of Industrial Engineers, 25 Technology Park/Atlanta, Norcross, GA 30092 (USA)
ISSN 0887-4719; Order No.IIE-P-293

L

[15]
Title: Development of a Real Time Scheduling System for Automated Production.
Author: Lipton, Michael J.
Corporate Source: ITP Boston Inc, Cambridge, MA, USA
Conference Title: Autofact '85 - Conference Proceedings.
Conference Location: Detroit, MI, USA Conference Date: 1985 Nov 4-7
Source: AUTOFACT, Conference Proceedings 1985. Publ by SME, Dearborn,MI,USA p13.45-58
Publication Year: 1985
ISBN: 0-87263-208-3
Abstract: Artificial intelligence, simulation, and heuristic techniques are used to plan and execute schedules in automated plants. The design, implementation and benefits of an automated production scheduling system are described. Examples are discussed and the pros and cons of the approach are summarized. (Edited author abstract)
Descriptors: *Production Control--*Scheduling; Management--Information Systems; Systems Science And Cybernetics--Heuristic Programming, Computer Aided Manufacturing--*Scheduling; Artificial Intelligence; Computer Simulation; Industrial Plants--Flexible Manufacturing Systems

Identifiers: Real-Time Scheduling System; Flexible Automation; Automated Production; Decision Support Systems; Install And Reconcile Schedule; Resources Planning And Allocation

[16]
Title: Proceedings of the 1992 Summer Computer Simulation Conference. Twenty-Fourth Annual Computer Simulation Conference
Editor(s): Luker, P.
Publisher: SCS, San Diego, CA, USA
Publication Date: 1992 Country of Publication: USA xxii+1273 pp.
Conference Date: 27-30 July 1992 Conference Location: Reno, NV, USA
Abstract: The following topics were dealt with: simulation methodologies; computer systems modelling; intelligent simulation environments; AI/KBS in simulation; communication and radar systems; physical/chemical/engineering applications; biomedical and environmental simulation; marine and undersea systems simulation.

[17]
Title: Conference on intelligent simulation environments
Edited by Luker, Paul A. Edited by Adelsberger, Heimo H.
Contributors: Luker, Paul A.; Adelsberger, Heimo H.
Publ: Society for Computer Simulation, San Diego, CA 1986, 167
San Diego, CA ; Society for Computer Simulation Jan. 23-25, 1986
Subfile: GCL (ACM Guide to Computing Literature) ACM
Descriptors: *I.6.1 -Computing Methodologies- Simulation And Modeling- Simulation Theory ; I.6.0 - Computing Methodologies- Simulation And Modeling - General

N

[18]
Title: Simulation In Material Flow Systems - Trends And Developments.
Author: Noche, Bernd
Corporate Source: Fraunhofer-Inst for Transport Engineering & Physics of Goods Distribution, Dortmund, West Ger
Conference Title: Record of Proceedings - The 19th Annual Simulation Symposium.
Conference Location: Tampa, FL, USA Conference Date: 1986 Mar 12-14
Source: Record of Proceedings - Annual Simulation Symposium 19th. Publ by IEEE, New York, NY, USA. Available from IEEE Service Cent (Cat n 86CH2286-3), Piscataway, NJ, USA p 11-41
Publication Year: 1986
ISSN: 0272-4715 ISBN: 0-8186-0715-7
Abstract: For simulating material flow systems, simulators and simulation packages are often used today which unfortunately do not represent the actual state of the art. The author examines this gap and discusses five areas for further development: integration of simulation into the general logistics planning; intelligent simulation packages; simulation and planning in service centers; research and development for planning of material flow systems; and real-time problems. It is concluded that future simulation developments will be deeply influenced by special applications like those for material flow systems. 20 refs.
Descriptors: *Materials Handling--*Computer Simulation; Computer Software --Applications

Identifiers: Material Flow Systems Simulation

O

[19]
Title: The role of artifical intelligence in discrete-event simulation
Artificial intelligence, simulation & modeling
Author(s): O'Keefe, R. M.
Contributors: Widman, Lawrence E.; Loparo, Kenneth A.; Nielsen, Norman R.
1989, 359-379 ISBN: 0-471-60599-9
John Wiley & Sons, Inc., New York, NY;
Price: $44.95
Subfile: GCL (ACM Guide to Computing Literature) ACM
Descriptors: *I.6.1 -Computing Methodologies- Simulation And Modeling- Simulation Theory- Types of simulation (continuous and discrete) ; G.3 - Mathematics of Computing- Probability And Statistics- Probabilistic Algorithms (Including Monte Carlo); I.2.0 -Computing Methodologies- Artificial Intelligence- General
Identifiers: Algorithms Performance Design Theory Experimentation Verification

P

[20]
Title: Supporting an intelligent simulation/modeling environment using the active KDL object-oriented database programming language.
Author: Potter, Walter D.; Miller, John A.; Kochut, Krys J.; Wood, Sanford W.
Corporate Source: Univ of Georgia, Athens, GA, USA
Conference Title: Proceedings of the Twenty-First Annual Pittsburgh Conference Part 4 (of 5)
Conference Location: Pittsburgh, PA, USA Conference Date: 1990 May 3-4
Source: Networks, Biomedical, Electronics, Hydrology, Data Bases and General Modeling and Simulation, Proceedings of the Annual Pittsburgh Conference v 21 pt part 4. Publ by Soc for Computer Simulation Int, San Diego, CA, USA. p 1895-1900
Publication Year: 1990

Abstract: This paper describes the design and prototype implementation of an object-oriented database programming language called Active KDL (Knowledge/Data Language). Active KDL is designed to support the complex information needs of engineering databases, expert databases, and intelligent information systems. In particular, Active KDL provides an integration of model bases, knowledge bases, and databases. The focus of this paper is on the evolution of Active KDL and on how Active KDL can be used to provide a very powerful intelligent modeling environment (we use a bank simulation example to show this). The inputs and outputs of models (e.g., simulation models, statistical models, financial models and decision models) can be stored in Active KDL since it supports complex objects. Active KDL also allows users to specify rules to capture heuristic knowledge and methods to specify behavior. Finally, Active KDL provides a simple mechanism for specifying concurrent execution, namely tasks embedded in active objects. These facilities provide a powerful mechanism for managing combined model, data, and knowledge bases. (Author abstract)

Descriptors: *DATABASE SYSTEMS

Identifiers: Intelligent Stimulation/Modeling Environment; KDL Object-Oriented Database Programming Language; Knowledge/Data Language (KDL); Active KDL; Object Oriented Database Programming

R

[21]

Title: Simulation convergence in a class of economic systems

Author(s): Racoveanu, N.

Journal: Economic Computation and Economic Cybernetics Studies and Research no.2 p.23-8

Publication Date: 1970 Country of Publication: Romania

ISSN: 0424-267X

Abstract: A convergence condition for systems belonging to class IMO pointing out that morphisms I to O make up a set, is dealt with. It is concluded that the systems theory and simulation are inseparable.

Descriptors: economic cybernetics; simulation

[22]

Title: Integrated intelligent simulation environment

Author(s): Ming Rao; Tsung-Shann Jiang; Tsai, J.J.-P.

Author Affiliation: Dept. of Chem. & Biochem. Eng., Rutgers the State Univ. of New Jersey, Piscataway, NJ, USA

Journal: Simulation vol.54, no.6 p.291-5

Publication Date: June 1990 Country of Publication: USA

ISSN: 0037-5497

Abstract: The authors describe a new simulation method for engineering applications. The architecture development is one of the most important techniques among the AI research tasks. An integrated intelligent simulation system is a large knowledge integration environment that consists of both symbolic reasoning systems (expert systems) and numerical computation packages. These software programs are controlled by a meta-system, which manages the selection, operation and communication of these programs. This new architecture can serve as a universal configuration to develop the fifth generation simulation environment for many complicated applications. (16 Refs)

Descriptors: digital simulation; expert systems; fifth generation systems ; integrated software; software packages

Identifiers: OPS5 application; simulation method; engineering applications; integrated intelligent simulation system; knowledge integration environment; symbolic reasoning systems; expert systems; numerical computation packages; meta-system; fifth generation simulation environment

[23]

Title: An intelligent simulation environment for optimal control systems

Author(s): Ming Rao; Tsung-Shann Jiang; Tsai, J.J.-P.; Chang, C.K.

Author Affiliation: Illinois Univ., Chicago, IL, USA

Conference Title: Proceedings of the 1987 Summer Computer Simulation Conference p.842-4

Editor(s): Chou, J.Q.B.

Publisher: SCS, San Diego, CA, USA

Publication Date: 1987 Country of Publication: USA xliv+1021 pp.

ISBN: 0 911801 20 0

Conference Date: 27-30 July 1987 Conference Location: Montreal, Que., Canada

Abstract: The authors illustrate the idea of applying AI techniques in simulation research and design to solve optimal control problems. The acquisition and utilization of expert knowledge for the domain problem are studied, and the problem-solving strategy is presented. An expert system for the simulation of optimal control problem-solving is developed, which has some important features: the whole search is divided into three reasoning levels such that the problem can be solved easily and search routine can be simplified; filter rules are utilized to reduce production rules and to enhance the program efficiency; the knowledge base can be modified and new rules can be created in production rule memory; certainty factor is used to represent the imprecise knowledge. (6 Refs)

Descriptors: control system analysis computing; control system CAD; digital simulation; expert systems; optimal control; optimal systems Identifiers: control system CAD; intelligent simulation environment; optimal control systems; AI; expert system; reasoning levels; filter rules; production rules; knowledge base

[24]

Title: Classifications of intelligent simulation systems

Author(s): Ming Rao; Yiqun Ying; Tsung-Shann Jiang; Psai, J.J.-P.

Author Affiliation: Dept. of Chem. Eng., Alberta Univ., Edmonton, Alta., Canada

Conference Title: Proceedings of the 1990 Summer Computer Simulation Conference p.700-2

Editor(s): Svrcek, B.; McRae, J.

Publisher: SCS, San Diego, CA, USA

Publication Date: 1990 Country of Publication: USA xix+1202 pp.

ISBN: 0 911801 74 X

Conference Date: 16-18 July 1990 Conference Location: Calgary, Alta., Canada

Abstract: The research and development of intelligent simulation systems have continued for several years, and its efforts have produced three types of software environment and architecture. They are: single expert systems that only process symbolic information and provide assistance to system engineers in decision-making process for off-line simulation and modelling; coupling systems that couple numerical computation programs into an expert system such that it can be used to solve engineering simulation problems; and integrated intelligent systems that are large intelligence integration environments, which can integrate different expert systems or numerical packages together to solve complex problems. (16 Refs)

Descriptors: digital simulation; engineering computing; expert systems; symbol manipulation; systems engineering

Identifiers: intelligent simulation systems; software environment; single expert systems; symbolic information; system engineers; decision-making process; off-line simulation; modelling; coupling systems; numerical computation programs; engineering simulation problems; integrated intelligent systems; large intelligence integration environments; numerical packages; complex problems

[25]

Title: Development of an expert system for intelligent simulation of housing modernization works

Author(s): Retik, A.; Marston, V.; Alshawi, M.

Author Affiliation: Dept. of Surveying, Salford Univ., UK

Conference Title: Artificial Intelligence and Civil Engineering p. 177-85

Editor(s): Topping, B.H.V.

Publisher: Civil-Comp Press, Edinburgh, UK

Publication Date: 1991 Country of Publication: UK vi+313 pp.

ISBN: 0 948749 14 8

Conference Date: 3-5 Sept. 1991 Conference Location: Oxford, UK

Abstract: A one year study is described which is investigating the feasibility of forecasting contract cost and duration by means of intelligent simulation. The intelligent simulation approach seeks to provide the user with probabilistic estimates generated by a production plan based model, and the ability to test the sensitivity of the outcome to factors such as different constraint conditions, resourcing levels or management strategies. The vehicle chosen for the study is multi-unit, low rise housing modernisation: these projects are characterised by the sensitivity of cost and time to operational factors, and by high levels of uncertainty. The authors outline the objectives of the work. They review the research methodology and focus upon the progress that has been made to date in the development of a prototype system. (15 Refs)

Descriptors: building; construction industry; digital simulation; expert systems

Identifiers: expert system; housing modernization works; contract cost; intelligent simulation; probabilistic estimates; production plan based model; constraint conditions; resourcing levels; management strategies; low rise housing modernisation; operational factors; research methodology

[26]

Title: An intelligent simulation environment for control system design

Author: Robinson, James T.

Author Affiliation: Oak Ridge Nat. Lab., TN, USA

Conference Title: 7th Power Plant Dynamics, Control and Testing Symposium Proceedings p.3.01-3.12 vol.1

Conference Title: Proceedings of the SCS Multiconference on AI and Simulation

Conference Location: Tampa, FL, USA Conference Date: 1989 Mar 28-31

Source: Simulation Series v 20 n 4. Publ by Soc for Computer Simulation Int, San Diego, CA, USA. p 28-31
Publication Year: 1989
ISSN: 0735-9276
Editor(s): Upadhyaya, B.R.; Katz, E.M.; Kerlin, T.W.
Publisher: Univ. Tennessee, Knoxville, TN, USA
Publication Date: 1989 Country of Publication: USA 3 vol. (x+1270+i+75) pp.
Conference Date: 15-17 May 1989 Conference Location: Knoxville, TN, USA
Abstract: The Oak Ridge National Laboratory is currently assisting in the development of advanced control systems for the next generation of nuclear power plants. The author presents a prototype interactive and intelligent simulation environment being developed to support this effort. The environment combines tools from the field of artificial intelligence; in particular object-oriented programming, a LISP programming environment and a direct manipulation user interface; with traditional numerical methods for simulating combined continuous/discrete processes. The resulting environment is highly interactive and easy to use. Models may be created and modified quickly through a window oriented direct manipulation interface. Models may be modified at any time, even as the simulation is running, and the results observed immediately via real-time graphics. (8 Refs)
Descriptors: artificial intelligence; control system CAD; digital simulation; engineering graphics; nuclear power stations; object-oriented programming; power station computer control; programming environments; user interfaces; Computer Simulation
Identifiers: USA; CAD; digital simulation; simulation environment; control system design; nuclear power plants; artificial intelligence; object-oriented programming; LISP programming environment; direct manipulation user interface; real-time graphics; Combined Continuous/Discrete Processes; Discrete-Event Simulations

[27]
Title: Record of Proceedings: the 22nd Annual Simulation Symposium.
Author: Rutan, Alan H. (Ed.)
Corporate Source: Raytheon Co, Radar Systems Lab, Wayland, MA, USA
Conference Title: Record of Proceedings: the 22nd Annual Simulation Symposium
Conference Location: Tampa, FL, USA Conference Date: 1989 Mar 28-31
Source: Rec Proc 22th Annu Simul Symp Record of Proceedings - Annual Simulation Symposium 22nd. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat n 89CH2740-9), Piscataway, NJ, USA. 193p
Publication Year: 1989
ISSN: 0272-4715
Abstract: This symposium proceedings contains 16 papers. The following topics are dealt with: distributed simulation; networks/distributed computing; multiprocessor/parallel-processor systems; **artificial intelligence/simulation synergism**; analog simulation; and languages/tools.

S

[28]
Title: Intelligent simulation systems for production prognosis and planning
Author(s): Shafranski, V.V.
Comput. Cent., Moscow, USSR

International Conference on Artificial Intelligence - Industrial Applications 9025005
Leningrad (USSR) 15-19 Apr 1990
Scientific Board for Artificial Intelligence of the Academy of Sciences of the USSR; International Federation for Information Processing
Prof. Ponomaryov, Artificial Intelligence Conference, Leningrad Institute for Informatics and Automation, 39 14th Line, Leningrad 199178, USSR

[29]
Title: Intelligent simulation environments: identification of the basics
Author(s): Snyder, J.; Mackulack, Gerald T.
Author Affiliation: Motorola Inc., Tempe, AZ, USA
Conference Title: 1988 Winter Simulation Conference Proceedings (IEEE Cat.#87CH2512-2) p.357-63
Conference Location: San Diego, CA, USA Conference Date: 1988 Dec 12-14
Editor(s): Abrams, M.A.; Haigh, P.L.; Comfort, J.C.
Publisher: SCS, San Diego, CA, USA
Publication Date: 1988 Country of Publication: USA xxi+896 pp.
ISSN: 0275-0708 ISBN: 0-911801-42-1
Source: Winter Simul Conf Proc 1988 Winter Simulation Conference Proceedings. Publ by IEEE, IEEE Service Center, Piscataway, NJ, USA. Available from IEEE Service Cent (cat n 88CH25122-2), Piscataway, NJ, USA. p 357-363
Conference Date: 12-14 Dec. 1988 Conference Location: San Diego, CA, USA
Abstract: A problem exists in efficiently combining a nondeterministic decision capability with a current discrete-event simulation language for use by the simulationist. The authors explore this problem in the context of the discrete-event simulation problem domain implemented in SIMAN. The purpose is (1) to provide an ontological definition of abstract ideas from data to wisdom, (2) identify a taxonomy of simulation and artificial intelligence combination dialects, and (3) establish the need for and then introduce a 'decide node,' which will assist the simulationist in incorporating a broader spectrum of the ontology more easily than current dialects allow. (25 Refs)
Descriptors: Artificial Intelligence; Digital Simulation; Computer Simulation--*Efficiency; Artificial Intelligence
Identifiers: Intelligent Simulation Environments; Nondeterministic Decision Capability; Discrete-Event Simulation Language; Siman; Abstract Ideas; Artificial Intelligence; Intelligent Simulation Environments; Ontology

[30]
Title: Specification of an architecture for intelligent simulation environments
Author(s): Suck-Chul Hong; Cochran, J.K.; Mackulak, G.T.
Author Affiliation: Syst. Simulation Lab., Manage. Syst. Eng., Arizona State Univ., Tempe, AZ.
Conference Title: Simulation and AI, 1989. Proceedings of the SCS Western Multiconference p.77-82
Editor(s): Webster, W.
Publisher: SCS, San Diego, CA, USA
Publication Date: 1989 Country of Publication: USA vii+138 pp.
ISBN: 0 911801 44 8
Conference Date: 4-6 Jan. 1989 Conference Location: San Diego, CA, USA

Abstract: Considerable discussion has appeared in the literature concerning intelligent simulation environments. Such environments are computer programs with both knowledge and computation components. Development of such software involves the combination of concepts and computer programming techniques from disparate sources. In addition, new methodologies for model creation, manipulation, and execution have been proposed. The authors present their views based on lessons from a previous intelligent simulation environment for the semiconductor industry. The critical new thrusts include generic models, user design goal knowledge bases, and many supporting data bases. These new ideas are presented in a unified format which they call a 'specification'. The implementation aspects in terms of hardware and software are also presented. This approach to preliminary specification appears to be a promising area for future research. (13 Refs)
Descriptors: digital simulation; formal specification; knowledge based systems; programming environments
Identifiers: intelligent simulation environments; computer programming techniques; model creation; semiconductor industry; generic models; user design goal knowledge bases; preliminary specification

W

[31]
Title: Artificial intelligence, simulation, and modeling
Author(s): Widman, Lawrence E. , Loparo, Kenneth A.
Interfaces, 1990, 20, 48- 66

Z

[32]
Title: Conceptual bases for artificial intelligence and simulation
Artificial intelligence, simulation & modeling
Author(s): Zhang, G. , Ziegler, B. P.
Contributors: Widman, Lawrence E.; Loparo, Kenneth A.; Nielsen, Norman R.
1989, 45-73
ISBN: 0-471-60599-9
John Wiley & Sons, Inc., New York, NY;
Price: $44.95
Subfile: GCL (ACM Guide to Computing Literature) ACM
Abstract Length: none
Descriptors: *I.2.4 -Computing Methodologies- Artificial Intelligence- Knowledge Representation Formalisms And Methods ; I.6.1 -Computing Methodologies- Simulation And Modeling- Simulation Theory; I.2.8 - Computing Methodologies- Artificial Intelligence- Problem Solving, Control Methods, And Search
Identifiers: Algorithms Verification Design Performance

Appendix C:

Source Code Listings: Definition Modules

```
{       Onur
        Started  : 931026
        Modified : 940309
}

DEFINITION MODULE AGVM;

FROM PartM   IMPORT PartObj;
FROM WCM     IMPORT WorkCenterObj;
FROM GrpMod IMPORT QueueObj;
FROM ResMod IMPORT ResourceObj;
FROM SystemM       IMPORT FMSystemObj;

FROM Animate       IMPORT DynImageObj;

FROM GTypes IMPORT PointType; {to save the translation}


FROM StatMod       IMPORT ITimedStatObj , TSINTEGER;
FROM Pie     IMPORT PiechartObj;
{for stats.}



TYPE

AGVState = (idle , FullMove , EmptyMove , WaitingMachine ,
WaitingTrack);


AGVObj = OBJECT;
      {PUBLIC}
            CurrentState : AGVState; {necessary to aggregate}
                  {TrackLeap.Happen saves the sate - needs to access!}
            ASK METHOD SetState (IN state : AGVState);
            TELL METHOD MoveToSideTrack;
            WAITFOR METHOD ComeBackFromSideTrack;
            image : DynImageObj;
                  {it should inherit from - but creating lots of
problems...}
                  {moved to public after TrackLeap accepts AGV, instead
of image.}
            ID : INTEGER;
            ASK METHOD SetSystemTo(IN sys:FMSystemObj);
            ASK METHOD BecomeUnavailable;
            isAvailable : BOOLEAN;
            Location : WorkCenterObj; {a pointer}
            TELL METHOD Serve (IN CustomerWC : WorkCenterObj);
            {ASK METHOD SetImage (IN img : ANYOBJ);}
      PRIVATE
            SaveTranslation : PointType;
            FMS : FMSystemObj;
            thePart : PartObj;
            Path : QueueObj; {que of TrackLEAP!}
```

```
                    {this does not need my CLONE(). ok not to be
OQueueObj}
            DestinationWC : WorkCenterObj;
            WAITFOR METHOD GotoDestination;
            WAITFOR METHOD TakeNextTrack;
      {OVERRIDE - if it had inherited from DynImageObj}
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {AGVObj};



AGVallocatorObj = OBJECT (ResourceObj);
      {PUBLIC}
            ASK METHOD ReportInitStats;
            ASK METHOD AdjustAggregateState (IN state : AGVState ; IN
displacement : INTEGER);
            ASK METHOD UpdateStats; {upon evaluation of batch}
            ASK METHOD SetSystemTo (IN sys:FMSystemObj);
            TELL METHOD SendAGVto (IN RequestingWC : WorkCenterObj);
            {INHERITED  Give (User,#desired);
                        TakeBack (User,#returned);
                        Transfer (from,to,#transfered);
            }
      PRIVATE
            AggregateState : ARRAY AGVState OF TSINTEGER ;{0-
NumberOfAGVs}
            AggregateInitState : ARRAY AGVState OF REAL;
                  {stats. during the beginning of the simulation run}
            AggregateBatchStateMonitor : ARRAY AGVState OF
ITimedStatObj;
                  {added as THE ONLY {a second} monitor to collect
stats. local to this batch}
            AggregateXAvgBatchState : ARRAY AGVState OF REAL;
                  {XAvg of the local stats.}
            InitTWMPie ,
            XAvgPie : PiechartObj; {pointers transferred in SetSystemTo}
            FMS : FMSystemObj; {need it to find the closest AGV}
            AGV : ARRAY INTEGER OF AGVObj;
      OVERRIDE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {AGVallocatorObj};


{END TYPE}
END MODULE.
```

```
{       Onur
        Started  : 931026
        Modified : 940326
}

DEFINITION MODULE AutoM;

FROM PartM   IMPORT PartObj;
FROM GrpMod IMPORT QueueObj,RankedObj;
FROM ListMod        IMPORT QueueList;
FROM WCM     IMPORT WorkCenterObj;
{@@@!FROM PartM    IMPORT NumKindsOfParts;}

FROM IOMod   IMPORT StreamObj , FileUseType(Output) ;

FROM Graph   IMPORT {to monitor the following into FMS.SimControlWindow}
             IDataPtMObj ,      {CurrentlyCollectingDecnBatchID}
             RDataPtMObj, RDataPt;   {XAvgTISofPartType[] , probab.s}

{the following stuff is to display the probab. input dialog}
FROM Window IMPORT WindowObj;
{FROM Graphic       IMPORT GraphicLibObj;}



TYPE


ProbArray = ARRAY INTEGER OF REAL; {RDataPt;}


XAvgArray = ARRAY INTEGER OF REAL; {RDataPt;}


AutomatonObj = OBJECT
      {PUBLIC}
            ASK METHOD SetNumChoicesTo(IN n:INTEGER);
                  {@constructor, had ObjInit accepted param.s}
            ASK METHOD SetLearningConstTo(IN a:REAL);
            ASK METHOD MakeANewDecn():INTEGER;
            ASK METHOD RewardItself(IN Choice:INTEGER; IN Beta:REAL);
            Probab : ProbArray;
                  {this has to be public to be able to report to file}

            ASK METHOD InputProbabs ( IN window:WindowObj;
                              IN prompt:STRING);
                  {called from the automaton monitor charts}
            ASK METHOD LoadFromFile (IN file : StreamObj);
                  {called from ANW.LoadFromFile}
            NumUpdates : INTEGER;
      PRIVATE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
            NumChoices : INTEGER;
```

```
            LearningConst : REAL;
            {Probab moved to public to be able to trace}
      END OBJECT {Automaton};




BatchRanker = OBJECT (RankedObj)
      OVERRIDE
            ASK METHOD Rank(IN a,b:ANYOBJ):INTEGER;
      END OBJECT;




AutomataNWObj = OBJECT
      {PUBLIC}
            ASK METHOD LoadFromFile (IN FileName : STRING);
            ASK METHOD SaveToFile (IN FileName, comment : STRING);
            ASK METHOD Receive(IN thePart : PartObj);
            ASK METHOD MakeNewDecisions();
            ASK METHOD CurrentDecision(IN fromWC,toWC :
WorkCenterObj):INTEGER;
            ASK METHOD RegisterAGVbrain(IN fromWC,toWC : WorkCenterObj);
{@}
{to monitor CurrentlyCollectingDecnBatchID and XAvgTISofPartType }
            ASK METHOD ActivateMonitors;
                  {can't do it in ObjInit - FMS seems to be unallocated
!!}

            Automaton : ARRAY[1..4],[1..4] OF AutomatonObj;
                  {this has to be public for the monitor charts to be
able to refer to these}

      PRIVATE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
            Decision : FIXED ARRAY[1..4],[1..4] OF INTEGER;
            CurrentBatch : QueueObj;
            FutureBatches : BatchRanker;
            CurrentBrains : QueueList;

            ASK METHOD EvaluateBatch;
            ASK METHOD CalcBatchBeta : REAL;
            ASK METHOD UpdateANWandXAvgs (IN BatchBeta:REAL);
            ASK METHOD UpdateRelatedAutomataWithPart (IN Beta:REAL;
INOUT{?} part:PartObj);
            ASK METHOD UpdateAutomataWithBrains (IN Beta : REAL);
            ASK METHOD ReportToFile (IN Beta : REAL);
            ASK METHOD ReportToSimControlWindow (IN Beta : REAL);
            ASK METHOD DumpFromFutureToCurrentBatch;

            CurrentlyCollectingDecnBatchID : LMONITORED INTEGER BY
IDataPtMObj;
```

```
            XAvgTISofPartType : XAvgArray;

            XAvgTISofPartTypeLOWERENV : XAvgArray;

            CumulTypeCnt : ARRAY INTEGER OF INTEGER;

            ReportFile : StreamObj; {to report Xavg.s, probab.s etc.}

      END OBJECT {AutomataNWObj};


BrainRecord = RECORD
      fromWCindex , toWCindex , DecisionMade : INTEGER;
      END RECORD;


{END TYPE}
END MODULE.
```

```
{       Onur
        Started  : 931022
        Modified : 940311
}

DEFINITION MODULE BuffM;

FROM GrpMod IMPORT QueueObj;
FROM SimMod IMPORT TriggerObj;
FROM PartM  IMPORT PartObj;
FROM Meter  IMPORT DigitalDisplayObj;
FROM Graph  IMPORT GraphVObj;
FROM Text   IMPORT TextObj;


FROM StatMod      IMPORT TSINTEGER , ITimedStatObj;

TYPE


BufferObj = OBJECT (QueueObj , GraphVObj)
{inherits from the QueueObj}
      {PUBLIC}
            ASK METHOD ReportToDisplays (IN MeanD , XAvgD :
DigitalDisplayObj);
            ASK METHOD ReportInitStats;
            ASK METHOD UpdateStats; {at batch evaluation}
            {inherited numberIn     : INTEGER;}
            {numberIn   : SINTEGER; - doesn't let me do this !}
            contents : TSINTEGER;
            InitTWM : REAL; {to report to file !}
            BatchContentsMonitor : ITimedStatObj;
            XAvgMeanContents : REAL;
                  {these two need to be public to report stats to
file...}
            WAITFOR METHOD BecomeAvailableAgain;
            ASK METHOD IsAvailable : BOOLEAN;
            ASK METHOD SetCapacityTo(IN capa : INTEGER);
            WAITFOR METHOD AddPart(IN NewPart : PartObj);
      PRIVATE
            ASK METHOD UpdateContents (IN value : INTEGER);
                  {I DO need the param.}
            capacity    : INTEGER;
            ContentsDisplay ,
            InitTWMDisplay ,
            XAvgTWMD : DigitalDisplayObj;
            name : TextObj;
            ReleasingPart     : TriggerObj;

      OVERRIDE
            ASK METHOD ObjInit; {QueObj does not have one, but
DigitalDisplayObj does !}
            ASK METHOD ObjTerminate;
            ASK METHOD Remove : {@PartObj}ANYOBJ;
```

```
            ASK {@this is the mask} METHOD Add(IN dummy : ANYOBJ);
            {THIS ASK ADD SHOULD NOT BE CALLED - IT'S THERE JUST TO MASK
ERRORS...}
END OBJECT {BufferObj};




InBufferObj = OBJECT (BufferObj)
{inherits from my BufferObj}
      PartArrived : TriggerObj;
      WAITFOR METHOD ReceivePart;
      OVERRIDE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
            WAITFOR METHOD AddPart(IN NewPart : PartObj);
END OBJECT {InBufferObj};



{END TYPE}
END MODULE.
```

```
{       Onur
        Started  : 940118
        Modified : 940309
}

DEFINITION MODULE MyControlM;


FROM Window IMPORT WindowObj;
FROM Graphic      IMPORT GraphicLibObj;
FROM Animate      IMPORT DynDClockObj;

FROM Menu   IMPORT MenuBarObj;
FROM Form   IMPORT DialogBoxObj;

FROM SystemM      IMPORT FMSystemObj;
{THE VARIABLE IS DECLARED HERE, to access stuff from menu}

FROM Meter  IMPORT DigitalDisplayObj { , DialObj} ;
      {for Timescale, batch# display}
FROM Chart  IMPORT ChartObj;
      {for XAvgTISDisplay, BetaDisplay and AutoMonitor[1..4,1..4] array}
FROM Pie    IMPORT PiechartObj;     {for PartsPercentDisplay, all the
stats pies}

FROM Text   IMPORT TextObj;
            {for labels in the WCAGV stats window}

FROM Image  IMPORT ImageObj; {for WC images}

FROM GTypes IMPORT ScrollDirectionType
(VerticalScroll,HorizontalScroll);


TYPE


myMenuBarObj = OBJECT(MenuBarObj);
      PRIVATE
            ASK METHOD ModifyTimescale;
            ASK METHOD HandleExit;
            ASK METHOD HandleLoad;
            ASK METHOD HandleSave;
      OVERRIDE
            ASK METHOD BeSelected;
      END OBJECT;


myClockObj = OBJECT (DynDClockObj);
      OVERRIDE ASK METHOD ObjInit;
      END OBJECT;
```

```
NonClosableWindowObj = OBJECT (WindowObj);
      OVERRIDE
            ASK METHOD BeClosed;
      END OBJECT;


SimControlWindowObj = OBJECT(NonClosableWindowObj);
      {PUBLIC}
            ASK METHOD LoadOtherGraphics;
                  {for pats TIS , parts pie , partStats}
            BatchNoDisplay , PartNoDisplay : DigitalDisplayObj;
            TScaleDisplay : DigitalDisplayObj;
            XAvgTISDisplay : ChartObj;
            PartsPercentDisplay : PiechartObj;
            BetaDisplay : ChartObj;
            PartInitTWMPie ,
            PartXAvgPie ,
            WCInitTWMPie ,
            WCXAvgPie  : ARRAY INTEGER OF PiechartObj;
            BufferInitTWMDisplay,
            BufferXAvgDisplay : ARRAY INTEGER,INTEGER OF
DigitalDisplayObj;
            BufferLabel : ARRAY INTEGER OF TextObj;
            AGVInitTWMPie ,
            AGVXAvgPie : PiechartObj;
            PartInitStatsWindow : NonClosableWindowObj; {menu has to
access this}
            PartStatsWindow : NonClosableWindowObj; {menu has to access
this}
            WCAGVStatsWindow : NonClosableWindowObj;
      PRIVATE
            ASK METHOD ConstructStatsWindows;
            SimControlMenu : myMenuBarObj;
            SimControlGraphLib : GraphicLibObj;
      OVERRIDE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {SimControlWindowObj};




SensitiveChartObj = OBJECT (ChartObj);
      {PUBLIC}
            ASK METHOD SetIDs (IN from,to : INTEGER);
      {the chart will call the proper automaton's input probab. routine}
      PRIVATE
            fromID , toID : INTEGER;
      OVERRIDE
            ASK METHOD BeSelected;
      END OBJECT {SensitiveChartObj} ;


AutoControlWindowsObj = OBJECT;
```

```
      {PUBLIC}
            AutoMonitor : ARRAY [1..4] , [1..4] OF SensitiveChartObj;
            CurrentDecnDisplay : ARRAY [1..4] , [1..4] OF
DigitalDisplayObj;
            ASK METHOD SetAnimMode (IN AnimOn : BOOLEAN);
      PRIVATE
            BlockWindow : ARRAY [1..4] OF NonClosableWindowObj;
            WCImage : ARRAY [1..4] OF ImageObj;
            AutoControlGraphLib : GraphicLibObj;
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {AutoControlWindowsObj};


{END TYPE}

VAR FMS : FMSystemObj;

END MODULE.
```

```
{       Onur
        Started  : 931030
        Modified : 931030
}

DEFINITION MODULE NewListMod;


FROM ListMod        IMPORT QueueList;


TYPE


OQueueList = OBJECT (QueueList)
        {not OVERRIDE}
        {PUBLIC}
        {       ASK METHOD ObjClone();
                what the heck will i do here -
                the manuals don't even talk abt. the param.s that are passes
into this !
                what is the old (original object's name ??? how can i refer
to it ???
        }

        ASK METHOD CLONE():OQueueList;
        {i'm assuming that modsim is capable of LATE BINDING
        so that i'll be able to pass OQueueList objects, as well...}

        END OBJECT {OQueueList};

{END TYPE}
END MODULE.
```

```
{      Onur
       Started  : 931007
       Modified : 940227
}

DEFINITION MODULE PartM;

FROM NewListMod    IMPORT OQueueList;
FROM ListMod       IMPORT QueueList;
FROM WCM     IMPORT WorkCenterObj;
FROM SystemM       IMPORT FMSystemObj;

FROM Image   IMPORT ImageObj;

{this is for the "reportive" PRec routine declaration}
FROM IOMod   IMPORT StreamObj;


FROM StatMod       IMPORT ITimedStatObj , TSINTEGER;



VAR    NumKindsOfParts : INTEGER;
{@@@   WARNING
       !!! IF YOU CHANGE THIS VALUE,
       go into the implementation module PartM
       and modify the CreateRoutes method od RouteContainerObj
       to reflect the changes...
       I KNOW THIS IS NOT THE BEST WAY OF DOING IT,
       BUT TEMPORARILY (UNTIL READ FROM A FILE)
       THIS WILL DO.
}


VAR
       {this cannot be a var. local to teh implementation module PartM,
       because the system has to NEW it and call SetSystemTo from it.}

       RouteHolder : RouteContainerObj;
             {this initializes and keeps the route patterns corresponding
             to different kinds of parts.}


VAR PercentArrivals : ARRAY INTEGER OF REAL;

CONST NumPartStates = 5;

TYPE

ProcessRec = RECORD
       WC     : WorkCenterObj;
       time   : REAL;
       END RECORD;
```

```
PathRec = RECORD
      fromWCindex : INTEGER;
      toWCindex : INTEGER;
      DecisionMade : INTEGER;
      END RECORD;



RouteContainerObj = OBJECT
      {PUBLIC}
            ASK METHOD SetSystemTo(IN sys:FMSystemObj);
            ASK METHOD CreateRoutes (IN PartsFileName : STRING);
            ASK METHOD IssueRoute(IN PartType:INTEGER):QueueList;
      PRIVATE
            FMS : FMSystemObj;
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
            route : ARRAY INTEGER OF OQueueList;
            ASK METHOD PRec(IN wc:INTEGER; IN time:REAL):ProcessRec;
            ASK METHOD ReportingPRec(IN file : StreamObj ; IN
wc:INTEGER; IN time:REAL):ProcessRec;
END OBJECT {RouteContainer};




PartState = (process , WaitForMach , WaitForAGV , AGVMove , AGVBlocked);

PartObj = OBJECT { @ (ImageObj) }
      {PUBLIC}
            State : ARRAY PartState OF TSINTEGER{0-1} ;
            ASK METHOD SetState (IN state : PartState);
            image : ImageObj;
            ASK METHOD PrepareFirstProcess();
            WAITFOR METHOD beProcessed();
            Type        : INTEGER;
            ReleaseTime : REAL;
            DecnBatchID : INTEGER;
            ASK METHOD NextDestination() : WorkCenterObj;
            ANWRoute    : QueueList; {que of PathRec's for automata
updating}
                  {this needs to be reached from
ANW.UpdateRelatedAutomataWith().}
            ASK METHOD RegisterPath (IN fromWC,toWC : WorkCenterObj; IN
DecisionTaken:INTEGER);
                  {can't WRITE to attrib.s even if they're public. -
needed from AGV.GotoDestination}
            TimeInSystem      : REAL;
            ASK METHOD SetTimeInSystem();
      PRIVATE
            AGVRoute    : QueueList; {que of ProcessRec's for AGV
routing}
            CurrentProcess    : ProcessRec;
            CreationCnt : INTEGER;
```

```
      {OVERRIDE - decided NOT to inherit for this one, too !!!}
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
END OBJECT {PartObj};

{END TYPE}
END MODULE.
```

```
{       Onur
        Started  : 931030
        Modified : 940321
}

DEFINITION MODULE SystemM;

FROM WCM     IMPORT WorkCenterObj {WCLoadUnloadObj};
FROM TrackM IMPORT TrackNWObj;
FROM AGVM    IMPORT AGVallocatorObj;
FROM AutoM   IMPORT AutomataNWObj;

FROM Window IMPORT WindowObj;
FROM Graphic       IMPORT GraphicLibObj;

FROM MyControlM    IMPORT      NonClosableWindowObj ,
                   SimControlWindowObj ,
                   AutoControlWindowsObj ;


VAR
      NumAGVs :INTEGER;
      AGVspeed : REAL;
      NumPalettes :INTEGER;
      DecnBatchSize :INTEGER;
      XAvgTISconst  : REAL; {used to update XAvgTISofPartType[]}
      XAvgConstStats : REAL ; {used to update all other statistics}
      GeneralLearningConst : REAL ; {= 0.005; {used to update probabilty
vectors by all automata}}
      UseLowerEnv : BOOLEAN; {autom}
      NoOfWarmUpBatches : INTEGER;
      WarmUpBatchesRemaining : INTEGER; {autom}
      ReptFileName : STRING;

      NumChoNeighbor : INTEGER;
      NumChoDiagonal : INTEGER;

      ProbabsAsBarCharts : BOOLEAN;
      TISProbabsMOD : INTEGER;

      SimplifiedVersion : BOOLEAN;
TYPE


FMSystemObj = OBJECT
      {PUBLIC}
            AnimWindow : NonClosableWindowObj;
            SimControlWindow : SimControlWindowObj;
            AutoControlWindows : AutoControlWindowsObj;
            AnimGraphLib : GraphicLibObj;
            ControlGraphLib : GraphicLibObj;

            TELL METHOD Go;
```

```
            WorkCenter   : ARRAY [1..4] OF WorkCenterObj;
            WC1,WC2,WC3,WC4 : WorkCenterObj;
                  {will point to the same ones, for debug purpoeses}
            TrackNW            : TrackNWObj;
            AGVallocator       : AGVallocatorObj;
            AutomataNW   : AutomataNWObj;
      PRIVATE
            ASK METHOD ObjInit;
            ASK METHOD InputPartsFileName (OUT PartsFileName : STRING);
            ASK METHOD InputReptFileName (OUT ReptFileName : STRING;
                                        IN PartsFileName : STRING);
            ASK METHOD ObjTerminate;
      END OBJECT {FMSystemObj};

{END TYPE}

VAR
      AnimOn : BOOLEAN;
      SimulationIsOver : BOOLEAN;

END MODULE.
```

```
{       Onur
        Started  : 931026
        Modified : 940307
}

DEFINITION MODULE TrackM;

FROM SimMod IMPORT TriggerObj;
FROM GrpMod IMPORT QueueObj;
FROM WCM    IMPORT WorkCenterObj;
FROM SystemM      IMPORT FMSystemObj;

FROM Image  IMPORT ImageObj;
FROM GTypes IMPORT  PointArrayType;
FROM AGVM   IMPORT AGVObj; {for the TrackLeap to Happen}

FROM IOMod  IMPORT StreamObj; {param. passed into LoadTrackList}

FROM GTypes IMPORT  PointArrayType;

FROM PartM  IMPORT PartObj;
        {so that Happen will modify the part's state properly..}


CONST NumTracks = 12;

{

         wc1                                  wc2
        (1)      ___ tr1____     (2)      ___ tr2___      (3)

         |           |            |
        tr7         tr8          tr9
         |           |            |

        (4)      ___ tr3____     (5)      ___ tr4____     (6)

         |           |            |
        tr10        tr11         tr12
         |           |            |

        (7)    ___tr5___    (8)     ___ tr6____     (9)
        wc4                                 wc3


NOTE : TrackLeaps are in
        (+ve) direction if going from a (lower#) junction to a (higher#)
junction.
        (-ve) direction o/w. (left, up --> negative.)
}


TYPE
```

```
TrackObj = OBJECT { (ImageObj);} ;
      {PUBLIC}
            ASK METHOD IsAvailableIn (IN dir : INTEGER) : BOOLEAN;
            image : ImageObj;
            ASK METHOD SetImage (IN i : ImageObj);
            WAITFOR METHOD BecomeAvailIn (IN dir : INTEGER);
            WAITFOR METHOD AllocateItselfIn (IN dir : INTEGER);
            ASK METHOD DeAllocateItselfIn (IN dir : INTEGER);
            TrackID : INTEGER;
      PRIVATE
            LastAllocationTime : REAL;
            CurrentDirection : INTEGER;
            idle : TriggerObj;
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {TrackObj};


TrackLeap = OBJECT
      {PUBLIC}
            ASK METHOD Init (IN aTrack:TrackObj ; IN
aDirection:INTEGER);
            WAITFOR METHOD Happen(IN AGV : AGVObj ; IN thePart :
PartObj);

            theTrack : TrackObj;
                  {this is to point to that track only ! NOT TO BE
NEW'ED !}
                  {I was planning to keep theTrack as private,
                   but RecalcEffectiveDistances uses it !}
            theDirection : INTEGER; {public for debugging purposes}
      PRIVATE
            ASK METHOD ObjTerminate;
            TrackImagePoints : PointArrayType;
      END OBJECT {TrackObj};




TrackNWObj = OBJECT (ImageObj);
      {PUBLIC}
            ASK METHOD SetSystemTo(IN sys:FMSystemObj);
            ASK METHOD EffectiveDistBtw (IN FromWC, ToWC :
WorkCenterObj) : REAL;
            ASK METHOD IssuePath(IN FromWC, TOwc : WorkCenterObj; INOUT
Decn:INTEGER) : QueueObj;
            ASK METHOD RecalcEffectiveDistances();
                  {AutomataNW will call this after making new decn.s}
      PRIVATE
            FMS : FMSystemObj;
            Track : ARRAY [1..NumTracks] OF TrackObj;
            AvailablePaths : ARRAY [1..4],[1..4] OF
                          ARRAY INTEGER OF {O}QueueObj;
```

```
                  {these paths need to be able to replicate themselves.}
                  {needs my .CLONE() method ! - use OQueueObj !}
                              {[fromWC,toWC][pathNo]=queOFtrackLEAPS!}
            ASK METHOD SetAvailPaths();
            ASK METHOD LoadTrackList(IN file : StreamObj;
                              IN cntFrom , cntTo , cntDecn : INTEGER );

            ASK METHOD TLObj(IN track:TrackObj; IN dir:INTEGER) :
TrackLeap;
            EffDistance : FIXED ARRAY [1..4],[1..4] OF REAL;
      OVERRIDE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {TrackNWObj};


{END TYPE}
END MODULE.
```

```
{     Onur
      Started  : 931026
      Modified : 940309
}

DEFINITION MODULE WCM;
{the WorkCenter Model}

FROM PartM  IMPORT PartObj;
FROM BuffM  IMPORT BufferObj, InBufferObj;
FROM SimMod IMPORT TriggerObj;
FROM ResMod IMPORT ResourceObj;
FROM SystemM      IMPORT FMSystemObj;

{FROM Animate     IMPORT DynImageObj;}
FROM Image  IMPORT ImageObj;

FROM StatMod      IMPORT ITimedStatObj , TSINTEGER;
FROM Pie    IMPORT PiechartObj;
{for stats.}

TYPE


WCState = ( idle , processing , blocked );


WorkCenterObj = OBJECT  {(ImageObj)}      ;
      {PUBLIC}
            ASK METHOD ReportInitStats;
            ASK METHOD UpdateStats;
            ASK METHOD SetSystemAndID(IN sys:FMSystemObj ; IN
id:INTEGER);
            WAITFOR METHOD Take(IN ArrivedPart : PartObj);
            ASK METHOD Give (OUT GoingPart : PartObj);
            ASK METHOD index ():INTEGER;
            TELL METHOD KeepBusy;
            image : {Dyn}ImageObj;
      PRIVATE
            ASK METHOD SetState (IN state : WCState);
            State : ARRAY WCState OF TSINTEGER{0-1} ;
            InitState : ARRAY WCState OF REAL;
                        {recorded after initReport for file report
during termination}
                  {stats. during the beginning of the simulation run}
                  {monitors automatically created}
            BatchStateMonitor : ARRAY WCState OF ITimedStatObj;
                  {second set of monitors, for stats. local to the
current batch}
            XAvgTWMState : ARRAY WCState OF REAL;
                  {XAvg of the local stats.}
            InitWCTWMeanPie : PiechartObj;
            WCXAvgTWMPie : PiechartObj;
            FMS : FMSystemObj;
```

```
            ID : INTEGER;
            thePart : PartObj;
            InputBuffer : InBufferObj;
            OutputBuffer : BufferObj;
      {OVERRIDE - not inheriting from ImageObj...}
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
      END OBJECT {WorkCenterObj};



WCLoadUnloadObj = OBJECT (WorkCenterObj)
      {PUBLIC} {NOTHING PUBLIC - GOOD !}
             {others will not know the difference!}
            CurrentlyReleasingBatchID : INTEGER; {has to be public for
the ANW to label the brains}
      PRIVATE
            { CumulReleaseCount : INTEGER; - no use}
            Alert : TriggerObj;
            PaletteBuffer : ResourceObj;
            WAITFOR METHOD ProcessAFinishedPart;
            WAITFOR METHOD ProcessARawPart;
            TELL METHOD beAlert1;
            TELL METHOD beAlert2;
      OVERRIDE
            ASK METHOD ObjInit;
            ASK METHOD ObjTerminate;
            TELL METHOD KeepBusy;
            ASK METHOD Give(OUT GoingPart : PartObj);
      END OBJECT {WorkCenterObj};

{END TYPE}
END MODULE.
```

```
MAIN MODULE Clever;

FROM SystemM      IMPORT FMSystemObj;
FROM SimMod IMPORT StartSimulation;
FROM Dynamic      IMPORT RealTimeAnimation;

FROM MyControlM IMPORT FMS;

BEGIN
      OUTPUT ("intelligent simulation model 940327 - INITIALIZING...");
      NEW(FMS);
      TELL FMS TO Go;
      RealTimeAnimation := TRUE;
      StartSimulation;
      DISPOSE(FMS);
END MODULE.
```

Appendix D:

Excerpts from Implementation Code: Examples of the Reduced Semantic Gap

```
{PUBLIC - AGVallocator Object}
TELL METHOD SendAGVto (IN RequestingWC : WorkCenterObj);
        VAR     ClosestAGV : AGVObj;
                cnt : INTEGER;
        BEGIN
        WAIT FOR SELF TO Give (RequestingWC,1); END WAIT;
                {to handle the # avail, request queueing, etc...}
        {find the first available one}
        cnt := 1;
        WHILE AGV[cnt].isAvailable = FALSE      INC(cnt);   END WHILE;
        {now cnt points to the first available one}
        ClosestAGV := AGV[cnt]; {temporary assignment - not necessarily true, of course...}
        WHILE cnt < NumAGVs             {if there's hope that we'll find others}
                INC(cnt);               {take a look at the next one}
                IF AGV[cnt].isAvailable                 {if this one is available,}
                AND     (  ASK FMS.TrackNW EffectiveDistBtw( AGV[cnt].Location , RequestingWC )
                        < ASK FMS.TrackNW EffectiveDistBtw( ClosestAGV.Location , RequestingWC)
                        )                               {and is closer than our candidate}
                {then}  ClosestAGV := AGV[cnt]          {...then update ClosestAGV pointer}
                        END IF;
                END WHILE;
        ASK ClosestAGV TO BecomeUnavailable; {has to do it here. see the note above.}
        TELL ClosestAGV TO Serve (RequestingWC);
        END METHOD;
```

```
{PUBLIC - AGV object}
TELL METHOD Serve (IN CustomerWC : WorkCenterObj);
         BEGIN
         {ASK SELF TO BecomeUnavailable;
         this is done from AGVallocator.SendAGVto.}
         DestinationWC := CustomerWC;
         SetState (EmptyMove);
         WAIT FOR SELF TO ComeBackFromSideTrack; END WAIT;
         WAIT FOR SELF TO GotoDestination; END WAIT;
         ASK CustomerWC TO Give(thePart);
{@31p}   ASK image TO AddGraphic(thePart .image);
         ASK thePart.image TO SetTranslation (0. , 1000.);
         ASK thePart .image TO Draw;
         DestinationWC := ASK thePart NextDestination;
         SetState (FullMove);
         WAIT FOR SELF TO GotoDestination; END WAIT;
         TELL SELF TO MoveToSideTrackWC;
{@31p}   ASK image TO RemoveThisGraphic(thePart .image);
                  {@@@graphic invisible until the WC Take's it ???}
         SetState (WaitingMachine);
         WAIT FOR DestinationWC TO Take(thePart); END WAIT;
         SetState (idle);
         ASK image TO Update;
         thePart := NILOBJ;
         isAvailable := TRUE;
         END METHOD;
```

```
{PRIVATE - AGV object}
WAITFOR METHOD GotoDestination;
{this routine is called both to send an empty AGV to the requesting WC
 and to send the loaded AGV to the next WC in the route.
}
         VAR CorrespondingDecn:INTEGER;
         BEGIN
         IF DestinationWC = Location RETURN; END IF;
                  {this can only be the case of AGV is empty.}
         Path := ASK FMS.TrackNW TO IssuePath (Location, DestinationWC, CorrespondingDecn);
                  {the AGV needs a path, whether it's empty or full}
                  {the var. CorrespondingDecn is used just for transfer purposes...}
                  {the part needs to register it for the AutomataNW to go thru, later.}
         IF thePart <> NILOBJ {then this is an AGV taking a part to its next route}
                  ASK thePart TO RegisterPath(Location, DestinationWC, CorrespondingDecn);
         ELSE {this is an empty AGV gong to respond a request}
                  ASK FMS.AutomataNW TO RegisterAGVbrain(Location , DestinationWC);
                  {brain asked to register before the agv starts the journey, so the decision is always the current one,
                  belonging to the batch currently being released.}
                  END IF;
         Location := NILOBJ;
         WHILE Path.numberIn > 0
                  {state was set to empty or full move in method Serve,
                  blockages, etc. will be handled in TrackLeap . Happen}
                  WAIT FOR SELF TO TakeNextTrack; END WAIT;
                  END WHILE;
         DISPOSE (Path);
         Location := DestinationWC;
         END METHOD;
```

```
{PUBLIC}
TELL METHOD KeepBusy {WorkCenterObj} ;
{for WC#'s 2,3,4 : instances of the parent class WorkCenterObject}
          BEGIN
          LOOP      {infinite - until the simulation is over}
                    IF InputBuffer.numberIn = 0
                              SetState (idle);
                              WAIT FOR InputBuffer TO ReceivePart; END WAIT;
                              END IF;
                    thePart := ASK InputBuffer TO Remove;
{@31p}              ASK image TO AddGraphic(thePart .image);
                    IF AnimOn
                              ASK thePart .image TO SetTranslation (0.,0.);
                              ASK thePart .image TO Draw;
                    END IF;
                    ASK thePart TO SetState (process);
                    SetState (processing);
                    WAIT FOR thePart TO beProcessed; END WAIT;
                    ASK thePart TO SetState (WaitForAGV);
                    SetState (blocked);
                    WAIT FOR OutputBuffer TO AddPart(thePart); END WAIT;
                    SetState (idle);
{@31p}              ASK image TO RemoveThisGraphic(thePart .image); {remove from the WC's image @ actually from the OB's !}
                    thePart := NILOBJ;
                    TELL FMS.AGVallocator TO SendAGVto(SELF) ; {@ IN 0.01;}
                              { IN ... : if there's an AGV that will become available after dropping its part off,
                                        make sure to grab that one instead of requesting anoter one.}
                    END LOOP
          END METHOD {KeepBusy};
```

```
{OVERRIDE - PUBLIC - WC#1 - load / unload }
TELL METHOD KeepBusy;
        BEGIN
        TELL SELF TO beAlert1;
        TELL SELF TO beAlert2;
        LOOP    {infinite - until the simulation is over - see teh last condition to ProcessARawPart}
                IF InputBuffer.numberIn > 0
                        WAIT FOR SELF TO ProcessAFinishedPart;
                        END WAIT;
                ELSIF (  (PaletteBuffer.Resources > 0)
                        AND  OutputBuffer.IsAvailable
                        AND NOT SimulationIsOver   )
                        WAIT FOR SELF TO ProcessARawPart;
                        END WAIT;
                ELSE
                        SetState(idle); {just to make sure...}
                        WAIT FOR Alert TO Fire;
                        END WAIT;
                        END IF;
                END LOOP;
        END METHOD {KeepBusy};
```

```
{PRIVATE - WC load/unload object}
WAITFOR METHOD ProcessARawPart;
        BEGIN
        WAIT FOR PaletteBuffer TO Give(SELF,1); END WAIT;
        NEW(thePart);
         {do now new before the transfer
                - for proper statistics collection}
{@31p}  ASK image TO AddGraphic(thePart .image);
        IF AnimOn ASK thePart .image TO Draw; END IF;
        ASK PaletteBuffer TO Transfer (SELF, thePart, 1);
        ASK thePart TO PrepareFirstProcess;
        ASK thePart TO SetState (process);
        SetState (processing);
        WAIT FOR thePart TO beProcessed; END WAIT;
        ASK thePart TO SetState (WaitForAGV);
        SetState (blocked);
        WAIT FOR OutputBuffer TO AddPart(thePart); END WAIT;
        SetState (idle);
        TELL FMS.AGVallocator TO SendAGVto(SELF);
{@31p}  ASK image TO RemoveThisGraphic(thePart .image);
        thePart := NILOBJ;
        END METHOD;
```

```
{PRIVATE - WC load / unload}
WAITFOR METHOD ProcessAFinishedPart;
        BEGIN
        thePart := ASK InputBuffer TO Remove;
{@31p}  ASK image TO AddGraphic(thePart .image);
        IF AnimOn ASK thePart .image TO Draw; END IF;
        ASK thePart TO SetState (process);
        SetState (processing);
        WAIT FOR thePart TO beProcessed; END WAIT;
        SetState (idle);
        {doesn't matter which state the part is in now...}
        ASK PaletteBuffer TO TakeBack(thePart,1);
{@31p}  ASK image TO RemoveThisGraphic(thePart .image);
        ASK FMS.AutomataNW TO Receive(thePart);
        thePart := NILOBJ;
        END METHOD;
```