Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

BUILDING FLEXIBLE AGV AND ASRS SYSTEM MODELS FOR FACILITY DESIGN PHASE APPLICATIONS

Ronald L. Webster
David F. Foster

Inland Steel Flat Products Company
3210 Watling Street
East Chicago, Indiana 46312

ABSTRACT

Simulation is an excellent tool to aid plant layout and design of automated material transport systems. However, when simulation is employed early in the design phase, the modeler faces the absence of a firm facility layout or operating logic. Indeed the model is normally used to test numerous designs and operating scenarios. The modeler can expect frequent changes to most operating parameters, and thus must develop a model which contains a maximum of flexibility. Additionally, oftentimes the types of material transporters to use have already been decided (e.g., AGVs and/or ASRS cranes), and the question is not so much is the system feasible, but rather what throughput can be achieved or sustained. In this case the model must contain sufficient detail to project overall system performance. These points can be further exacerbated by a compressed facility design schedule.

This paper reviews, in general, important input, output, and model integration factors to consider when developing detailed flexible models of AGV and ASRS systems. Also, examples are presented for setting-up a model using the SIMAN simulation language, with FORTRAN enhancements.

1. INTRODUCTION

The utilization of simulation early in the design or feasibility study phase of a new facility construction project obviously affords the best opportunity for realizing significant benefits. However, it also provides an opportunity for frustration on the part of the modeler and the client due to the dynamic nature of the environment during that period. Initially it is often difficult to define the system to be modeled in sufficient detail to satisfy the modeler. The entire process is iterative wherein early on the client reacts to interim model design or results in a manner such as "I'm not sure what I want, but that's not it!".

If the situation is one of constrained resources, the client may be responsible for several activities during the facility design. He or she may be responsible for defining production line and material transport system specifications prior to issuing bid notices, evaluating equipment manufacturer's quotes, developing manufacturing processing costs, and visiting plants and vendors to inspect alternative operations. Many times everything is further compounded by a severe time limit. In such cases the modeler must be both efficient and flexible, and so must the model under development.

An early consideration in the planning of any model is the level of detail necessary to meet the study objectives. In an example of a model which is developed primarily to address material flow questions, if the types of material transport systems have already been chosen, and perhaps even their manufacturers, the feasibility of using alternative systems does not need to be tested, but rather the ultimate capabilities of the transport systems must be determined. In this scenario, a model of significant detail is required in order to predict maximum sustainable system throughput. Thus it is in the best interests of the modeler to build in as much detail as possible from the start without, however, sacrificing model flexibility.

When the material transport systems include AGVs and ASRS cranes, numerous factors can be anticipated by the modeler without significant input from the client. These options can be planned for and built into the model early in order to prevent future extended coding delays as the overall facility design evolves. Certain of these factors involve simply providing changeable user variables to represent such things as AGV velocity or ASRS deposit times. Others require more complexity, such as the number of aisles an ASRS crane will service. Providing flexibility early is worth the effort since virtually any modification late in model development can be extremely time consuming when verification and validation effort is included.

The remainder of this paper discusses key factors governing AGV and ASRS material transport system design and operation. Examples are presented of flexible model input utilizing SIMAN and FORTRAN, and the integration of the factors into the model is discussed. Important model output to facilitate the analysis of a transport system design, and examples of standard and expanded output are also reviewed.

2. SIMULATION OBJECTIVES

Unless the client is familiar with simulation, he has a difficult time defining specifically what the requirements are in terms of design and output. He may answer the simulationist's question by simply directing the question back to the modeler. "I don't know. What do you normally do?" Thus the modeler must provide certain expertise and ask more specific questions. For material flow models one can anticipate the pertinent objectives of the project to include areas such as:

. How many transporters are required to sustain production levels?
. How much interference time will the AGVs and ASRS cranes experience?
. Where should the pick-up and delivery points be located?
. What ASRS storage capacity is required?
. What is the AGV or ASRS system maximum throughput capability?
. What is the risk of having a system that can handle less than peak production throughput?

The challenge when initially constructing a model early in the facility design phase is to develop one that can flexibly analyze numerous scenario variations.

3. CONCEPTUALIZATION AND DESIGN

More than just a benefit, flexibility is a concept — its principles intertwined with other concepts at the time of their definition, before actual design and development. In terms of AGVs and ASRS, an understanding of the physical systems must preceed any concept of their simulation: flexibility is incorporated simultaneously when structuring those concepts.

The movement theory of AGVs and ASRS cranes are definitely understood topics: thoroughly discussed and extensively modeled. However, an important point is that when creating such simulation software (not desirous of relying on vendor models, sometimes based on costly networking philosophies and void of AGV/ASRS interface), an abstract focus on how these vehicles ought to be regulated is critical. In fact, probably the most important aspect is ignoring (temporarily) the practical factors that tend to constrain pure definition. It is after such a conceptualization that the limits of actual need should enter, because then a true perspective of the degree of compromise required by the situation can be gauged.

The basic concepts that were arrived at are 1) AGV movement decisions can be regulated by a few general rules, criteria similarly used by humans when moving about. To enable this, 2) AGV intentions including remaining route segments, move time to any node, load status, etc., must be available to the entire system. In addition, 3) clusters of track proximate to the intended route segment should be reservable by any AGV (to preclude entry by another transporter), which also implies an open-status track system. Finally, 4) pickup and retrieve requests should be stored and made available for later analysis by AGVs or cranes searching for work. The implications of the above are that multiple types of transporters can be regulated concurrently by the same software when common status, track, and routing conventions are used. These design criteria should conceptually be able to support AGVs, ASRS cranes, carts and overhead cranes all resolving common interferences although normally moving in different planes. As a side note, conflicts between overhead bridge cranes on a common runway, although not in this example, are extensible by simply inserting route segments that accommodate their unforeseen backup moves. Of concern at this stage is the potential CPU-burden of processing such concepts.

Besides those affecting transporters in general, one additional concept for flexible ASRS modeling is that the number of warehouses, their layout, types of product flowing through, initial inventories, and rules optimizing crane movements are all parameters that can be easily adjusted through a few numbers (in the SIMAN experimental file). However, this ease of use depends on one aspect: fast, flexible adjustment and operation of a single array tracking specific bins, by product-type in the inventory system. Careful design of this array greatly improves set-up time and allows automated change of most of the important factors in sizing and predicting ASRS throughput, which are so important later in making the many simulated runs needed for a comprehensive perspective.

After considering the above concepts and their probable development overhead against the facility's likely demands in material and traffic densities, frequent design changes, and probable volume of proposals and analyses, it was clear that a precise tracking of movements was crucial to the joint buy-in of our foreign and domestic clients. We decided that compromise early on would affect our later ability to give clear analysis in an acceptable timeframe. Consequently, at this point, the precision capabilities of the system; routing, track, status, and interference mechanisms is a function of the flexibility of the chosen simulation language. In SIMAN, the overall command structure and availability of FORTRAN/C-level tools are sufficient for design of a precise and flexible AGV simulation system.

Within the software structure, the afore-mentioned concepts result in flexibility in many areas. This section will elaborate within the following subsystems: open-status of transporters and track; task selection; AGV and ASRS crane movement dynamics; and ASRS sizing and inventory operation.

3.1 AGV and Track Status

Underlying the software that selects AGV work and regulates their travel are the subsystems that define AGVs and track, and maintain their status. It is critical that this information be accessible to all transporter entities in the model at all times. Another key part of molding the AGV system into a generic "engine" is the indexibility of the data access routines. Consequently, it is important that the data structure be open to both language blocks and FORTRAN code (we use SIMAN Parameter sets), writable to enable re-definition, and organized for indexed, bi-directional access. Our arrays are arranged by: AGV definition and status, track segment definition and status, route definition, and interference set definition. Figure 1 illustrates some typical array calls in a few instances. Of some note are the AGV "current route number" and the "interference set number" fields which effectively link AGV status with track information. In addition, "time started segment" in the AGV array when used with track "segment length" brings time-projection to the scope of analyses that AGVs can perform. These data structures as well as others whose design is discussed, are further amplified in implementation sections later in the paper.

Situation	Known Field	Array and List of Accessed Fields
Remaining time to complete current segment	AGV	A AGV A Time Start Segment A Segment T Segment Length
How far ahead is AGV's route open	AGV	A AGV A AGV Route R Route R Segments List T Segment T AGV Claiming Segment
Block others from a segment and from related segments	Segment	A Segment T Interference Set I Segments List

A AGV Status Array
T Track Status Array
R Route Definition Array
I Interference Set Array

Figure 1. Examples of Status Array Access

3.2 Task Selection

Another ingredient to flexible AGV software is the storage and analysis of work requests. Instead of, for example, checking status of all pick-up queues in its domain (and burdening the CPU) the AGV can automatically answer the next chronological request. Or in some instances, choose out-of-sequence for greater efficiency. In an abstract sense, open arrays containing work requests are channels for making a planned facility more efficient, because they allow simulation to originate the rules it will actually use. Simulation, then, is able to prove or improve throughput, before guaranteed equipment contracts are signed.

Again, we use open-accessed arrays, which prove very flexible in the modeling cases where data is needed by some unsuspected source. In such instances, use of closed attributes or non-global medium tend to build networks of temporary, disjointed variables.

Our needs for storing AGV requests are relatively simple, merely a chronological listing of the originating station number. ASRS requires a somewhat more complex arrangement because of the need for keeping requests during the dual moves (empty and full) in executing requests. The ASRS software uses one "mailbox" per warehouse; AGVs require one for each facility or general area.

3.3 AGV Movement Dynamics

The system regulating transporter movements is the heart of our model. Flexible AGV/ASRS crane interface, utilization and blocking statistics, battery drain and charge dynamics, and de/acceleration emulation are some of the spin-offs made easier because of the system's design.

The concept that AGV movements can be regulated by an "engine" containing few, constant rules depends on each AGV knowing the course that others will take. In this method, given a moving AGV, the detection of interference depends on other AGVs scanning their entire route in advance, claiming open segments until conceding one because of unresolved conflict. An AGV, when encountering a claimed segment is then able to intelligently decide through the logic system whether its priority, destination, computed time to mutual collision, or other criteria will enable over-riding the prior claim. Especially applicable for AGV/ASRS interface and complex track layouts, is the concept of an AGV blocking others from a predefined area of many track segments, defined as an interference set. Other AGVs are then able to recognize infringement on a segment not listed in their routes.

Also important is the need to systematize the queueing and release of blocked AGVs. Flexibility of this depends on AGVs signaling the system as each station is passed (our signal is the station number). A blocked AGV can then be sent to a queue that is defined relative to its station. Before being queued, the next station of the oncoming AGV can be computed, which is the releasing signal when that station is passed.

Such a methodology, although CPU-intensive, is not excessive because open segments are automatically claimed and interference resolution is fast because of relatively few rules.

The algorithm regulating interference must be as generic as possible: member rules proceed from general to specific, filtering away resolved conflicts and allowing the more complex to continue. The last equation compares mutual travel times to the point in question, deciding if the difference falls outside a specified time (we use five AGV-lengths). Appendix A traces the logic through the sequential steps.

A situation requiring design provision is the one shown in Figure 2. A subject AGV (sagv), is at a junction, evaluating whether to proceed left, a path which will ultimately gridlock an oncoming AGV (oagv). The algorithm allows sagv access to segments 14 and 15 because oagv is more than 5 agv-lengths from those points (assumed), but oagv's claim to segment 16 is upheld. To resolve this, the software temporarily stores contested segments allocated to sagv during its route analysis, along with the conflicting oagv number. Consequently, segments 15 and 14 are returned to oagv, who matches the stored oagv number; and sagv is blocked on segment 13.

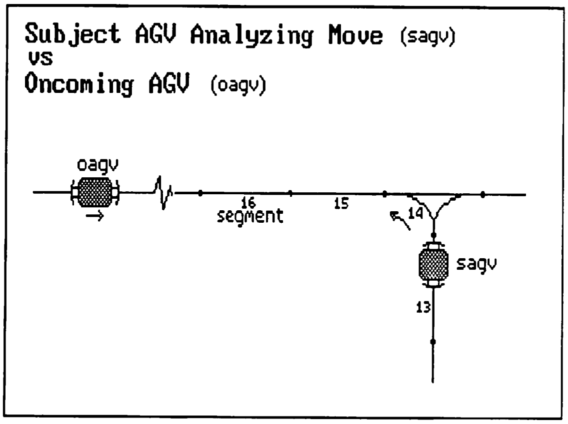

Figure 2. Potential AGV Gridlock

3.4 ASRS Inventory Operation and Sizing

Flexibility is designed into the ASRS system primarily through tracking of inventory. In order to gauge ASRS size and throughput, mimicing its efficiency logic is necessary, which requires tracking of specific bins and product-types. To accommodate this, an array is needed to quickly access bin information from knowledge of either product-type for retrieval or bin number for storage. The array also must easily adjust to new product types and quick change of ASRS dimensions for testing design alternatives.

In structuring the array it is necessary to accommodate the methodologies of ASRS operation. For example, when picking up a load for storage, an empty bin must be found. That bin must be marked as "claimed" until the crane arrives, in order to preclude its use for other reasons. Similarly, the retrieval subsystem must temporarily reserve chosen inventory.

For our purposes, it is unnecessary to track specific pieces through the inventory system; it is sufficient for our sizing and throughput analysis to randomly pick unique bins from lists of specific product types. The method considers empty bins a product type; consequently, for normal single-cycle ASRS operation, the array's dual pointer system allows consistent two-read choosing of empty or full bins.

In order to increase efficiency, actual ASRS software uses algorithms that choose storage locations which reduce move times to future work. When choosing an empty bin, for example, the system already knows the location of its next retrieve. It therefore chooses the empty (closest) bin which will minimize its future travel to that retrieve. Our array is designed to accommodate choosing of empty bins based on prior knowledge of full bin numbers. This choice uses predefined search patterns around the target full bin, checking the array (one-record reads) until an empty bin is found. Appendix B illustrates the design of the array and its operating procedures.

The design of the inventory array allows for another opportunity in flexibility: rapid adjustment of ASRS dimensions. We often need to experiment with different designs of ASRS bins, rows, columns, etc.; as well as the number of warehouses themselves and their product initialization levels. Open-access arrays can again be used (SIMAN parameter sets) during array creation and initialization to specify relevant parameters. Offsets distinguishing division between warehouses in the array are automatically computed, written during initialization, and used throughout the model run.

4. AGV SYSTEM VARIABLES

One potential advantage of any automated material transport system is its predictable or standard operating practice. Since there is no human element involved in the operation, certain cycle times and operating logic can be treated essentially as constants. For example in the case of AGVs, travel time for a cycle consists of empty travel time to pick-up a load (a function of acceleration, velocity, deceleration, and distance traveled which will be consistent for a particular move), pick-up time (a constant for a given load type), loaded travel time, any on-board process delays, and deposit time.

There are both static and dynamic parameters associated with AGV systems. The static parameters include the general equipment specifications – acceleration, empty and loaded velocity (these may differ for curved and straight path segments), deceleration, pick-up and deposit times, and process control communication time (in large systems this factor may need to be represented by a distribution since message transfer time is a function of the load on the communication system). These factors are usually relevant only to each individual AGV, and they are not time dependent.

Other key parameters are dynamic; they are time dependent and some methodology must exist that allows each AGV to check on the parameter status of all other transporters. These dynamic parameters include AGV location within the system, AGV travel route, empty/loaded status, the priority of the current move, and the number and priority of moves waiting for an AGV response.

Lastly, the AGV paths or routes must be defined in terms of segments (start point, length, and endpoint), interference points for intersecting or bi-directional segments, and priorities at interference points.

5. AGV SYSTEM PARAMETER SET-UP

Static parameter assignments are made through standard variables. Although the variables normally do not change during an individual simulation run, they may vary from run to run throughout project analysis. Figure 3 shows an example of an AGV static parameter segment of a SIMAN initialization file; U-type (user function) variables are chosen because statistical output data is not required for the variables and they conserve memory.

```
;      AGV Equipment Specifications
;
U011   60.98   Empty Maximum velocity (mpm)
U012   60.98   Loaded Maximum velocity (mpm)
U013   24.40   Slow move velocity (mpm)
U014   12.20   Creep move velocity (mpm)
U015   0.510   Empty accel rate (mps2)
U016   0.510   Loaded accel rate (mps2)
U017   0.510   Empty decel rate (mps2)
U018   0.510   Loaded decel rate (mps2)
U019   0.267   Pick-up time (min)
U020   0.267   Deposit time (min)
U021   0.050   Communication time (min)
```

Figure 3. AGV Static Parameters

Dynamic parameters can be handled through standard variable assignments also, but using an array has advantages. Each AGV has its own data set that includes all parameters which change frequently during a simulation run. The ability exists to read and write values from both the simulation language and the base language (FORTRAN for SIMAN) enhancements. In the case of SIMAN, user defined parameter sets are utilized. The values are effectively global; other transporters or system operations can check the status of these parameters at any time. The values could be carried with the AGV as entity attributes, but the use of parameter sets conserves memory, which is a concern in large models, and minimizes potential attribute conflicts at AGV/load interface points. Figure 4 shows the AGV dynamic parameter set-up in the experimental frame of a SIMAN model.

```
Parameters:
 ;      Seg Trv        Mv  Trav           Int
 ; Seg  Dst Dir Rt  #  Pr  Time    St Mt  Set Typ

 1,  -47, 28,  1,  7, 1,  1, 0.53,  1,  0,   0,  0:
 2,  244, 23,  1, 15, 1,  1, 0.71,  0,  1,   0,  0:
 3,   97, 73,  1, 34, 2,  1, 0.81,  0,  1,  76,  1:
 4,    0,  0,  0,  0, 0,  0, 0.00,  0,  0,   0,  0:
 5,    0,  0,  0,  0, 0,  0, 0.00,  0,  0,   0,  0:
```

Figure 4. AGV Dynamic Parameters

Each array provides AGV status regarding: 1) current path segment the AGV is traversing, 2) endpoint (destination station) on the segment, 3) direction of travel, 4) the route number which contains all the segments for the current assignment, 5) segment count number, 6) the priority of the move, 7) time to traverse the segment, 8) blocked, at standby, end of route flag, 9) empty/loaded flag, 10) interference set for the segment, and 11) type of AGV in a multi-type system. Additional elements can easily be added if other factors are pertinent. Of note is the negative segment number, which means that the AGV is moving in the direction opposite the segment's definition.

In the model itself the AGV is treated as an active entity. Each AGV has its own parameter set, and the number of parameter sets (or AGVs) is potentially unrestricted. In the above example there is room for two more AGVs should analysis of interim simulation results necessitate additions.

The total AGV system path layout is broken into appropriate segments. Segment endpoints consider intersections, pick-up and deposit points, and whether the segment is uni- or bi-directional. AGV segments are defined in experimental frame parameter sets also, since several elements are needed to define a segment (see in Figure 5).

```
Parameters:
;               AGV Track Segments
;
;
;      Seizd  Station            Queue
;Trk    by     Nums    Lngth     Nums     Intfce
;Resc  AGV   Str End  (meters) Str  End    Set#
;
 68,    0.,   7.,  8.,   6.0,   11.,  17.,   63.:
 69,    2.,   9., 12.,   8.8,   10. , 25.,  297.:
 70,    0.,   8., 39.,   3.8,   17.,  24.,  219.:
```

Figure 5. AGV Path Segment Definition

These arrays provide the AGV entity with travel information: 1) identifying segment (parameter set) number, 2) the AGV claiming the segment, 3) start and 4) endpoint (station) locations, 5) length of the segment, 6 & 7) wait points (queues) when interference is encountered or the AGV has completed all activity and is awaiting its next assignment, and 8) a table number identifying an interference set which contains the numbers of all intersecting segments. As track layout evolves, parameter sets are added, deleted, or modified as necessary. Of note is the correlation between the direction-of-travel field in Figure 4 with the station numbers in Figure 5. The direction number, when added to the first listed station number, gives the AGV's starting station for its direction of travel. This method facilitates use of bi-directional segments.

The total path, or route, an AGV will follow to complete a pick-up or delivery is defined in read only arrays; individual paths are not updated during the simulation run. Therefore, in this example, SIMAN table sets are used. The route that the AGV needs to travel is chosen within the model FORTRAN code. The set itself contains an identification number (the table set number), how many segments in the total route, and the individual segment numbers which constitute the route.

A system to prevent AGV collisions is included. This example uses an interference set concept. Each set is a table identifying all the segments that must be "free" before the AGV is allowed access to the desired segment.

6. ASRS SYSTEM VARIABLES

Although ASRS systems are automated operations and thus also ease modeling efforts due to their predictability, they offer challenges different from AGVs. The cranes routinely move through three dimensions, they do not interfere with other ASRS cranes, but they do interface with the entry and exit material transport systems, and that interference is of primary concern.

The ASRS cranes have the same series of static travel time related parameters as AGVs (i.e., acceleration, maximum velocities, deceleration, communication time, etc.), however they have a set for each direction of travel — bridge, trolley, and hoist — as well as mast rotation. These parameters remain constant throughout a simulation run, and often from scenario to scenario for a particular system. Within the model coding, algorithms are included that 1)calculate travel time in each direction and 2)consider simultaneous travel in more than one direction.

During simulation runs, when ASRS inventory information is accessed by algorithms optimizing crane movements, or computing their move times, there is a frequent need to convert bin numbers to their four coordinates. To accommodate this and improve execution speed, special conversion equations are used to obtain either the bin number or its coordinates directly from knowledge of the other.

Additionally, ASRS systems have static parameters describing the general physical configuration of the storage system which frequently are varied throughout the course of a facility design. These parameters include the number of cranes, aisleways, storage bins (horizontally and vertically), the aisle widths, the coordinates of pick-up and deposit stations and the dimensions of each bin opening.

Even though the ASRS cranes do not interfere with one another in this example, a set of interference parameters similar to those described for AGVs is provided. The ASRS system interacts with other material transport systems, and it is usually necessary for one system to check the status of the other before proceeding in interface areas.

Load storage is segregated by product type; variables representing the distribution of product to the individual ASRS bays are provided for the delivery transporters. Finally the routes that ASRS cranes follow are also described. These routes are generally less complex, however, than those for AGVs due to the aisleway restrictions.

7. ASRS SYSTEM PARAMETER SET-UP

The static parameters representing ASRS crane specifications are set-up in the manner described for AGVs, that is through the use of standard variables defined in an initialization file to expedite changes. Variables are required for the bridge, trolley, and hoist movements. Speed adjustment factors are also necessary because the maximum speed an ASRS crane will achieve for a particular movement is a function of the total distance it will traverse in a given direction. This variable maximum velocity is then utilized in the algorithm to calculate crane move time.

An additional set of static parameters is unique to the ASRS system, the bay configuration specification. Each bay in the ASRS system is configured differently in terms of the number of rows and columns per rack. It is convenient to arrange the configuration data in an array to facilitate understanding, runtime updating, and conervation of variables and memory. Figure 6 details a sample ASRS bay layout and the pertinent array elements. Updated counts for all inventory subsets within the ASRS bay are also stored in the array.

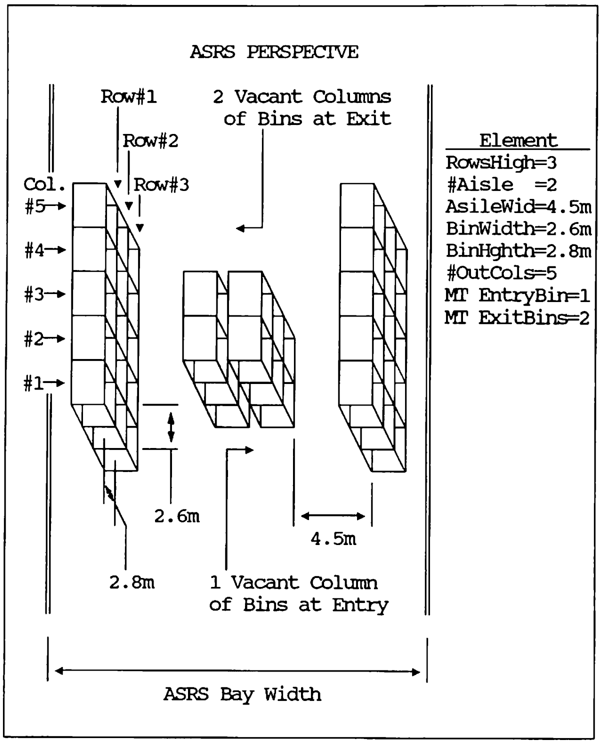

Figure 6. ASRS Physical Layout

ASRS path segments are defined in arrays (in a fashion comparable to AGVs) as shown in Figure 7. The stations, queues, and interference sets have the same meanings as those in AGV segment parameter sets. The x, y, and z values define the endpoint locations in three dimensions, and ø the crane ram orientation. The move time is calculated in model algorithms and then written into the array for each individual move.

```
;                  ASRS Crane Track Segments
;
;      Note: z Coordinate must be negative (-) for all segments
;         whose travel time is adjustable.
;
;
;                                          Start  Sta'n   End Stat'n
; Seiz  Statn   Move  Que#                 Orientation  Orientation
;   by  Nums   Time Str Ed  Intfc           Colmn(2)    Colmn(3)
;  Crn Str Ed  (min)          Set#         x    y  z   ø  x    y    z  ø
;
76,  1., 23.,  6.,   1.0,  23.,  89.,  0.,    0,   0, -1,  0,  0,    90,1.3, 0:
77,  0., 18.,41.,   1.2,  29.,  41.,  0.,  4.5, 5.4,  1, 90,  0,  10.4, 1, 90:
78,  0. ,43.,42.,   1.1,  72.,  42.,  0.,    0, 5.4,  1, 90,  0,  10.4, 1, 90:
```

Figure 7. ASRS Path Segment Definition

Interference sets and routes for ASRS crane travel are structured and utilized in the same manner previously described.

8. GENERAL MODEL OUTPUT REQUIREMENTS

Planning for output is also an important factor in model design. Early definition of what performance measures are key for the material flow systems allows efficient programming, saves time in the long run, and again demonstrates the importance of defining the objectives and client requirements for the model. With AGV and ASRS systems, numerous indicators should be monitored, some perhaps are not very obvious however. The objectives of the project determine the relative importance of each indicator.

There are a minimum of three general categories of output to consider: 1) counters, 2) descriptive statistics (i.e., average, maximum and minimum value, and a measure of variation), and 3) the trend of the statistic over time. Simulation packages normally provide a way to obtain counters and descriptive statistics. The package may even provide routines for obtaining the trend data (e.g., SIMAN's output processor), but the number that can be obtained for a single simulation run are severely constrained by the operating system under which the simulation is being executed. Thus getting trend data, especially if many (more than 10) plots are required necessitates additional coding effort.

9. AGV OUTPUT STATISTICS

For individual AGVs, potentially the only counters necessary are the number of loads moved (for each load type available) and the number of loads that required moving. The latter is needed since it is usually necessary to compare AGV system capability and required system throughput.

In terms of discrete time persistent statistics, it is important to capture the AGV's utilization (or non-charging time for battery powered vehicles), charging, and interference time. Utilization rate is an obvious statistic for any system. However, DC powered vehicles such as AGVs that use an opportunity charging concept have a maximum long term utilization limit of 45%-55% unless special batteries or charging mechanisms are employed. Thus the short term and cumulative charging time must be tracked. (Actually consideration should be given to measuring subsets of utilization such as travel empty, pick-up, travel loaded and deposit times since they provide added information regarding overall system efficiency. Battery discharge rates also differ for each subset.)

In order to capture trend data, statistics generated at predefined time intervals (e.g., number of loads, percent utilization, etc.), for each counter and time related statistic output, special coding is required. The plot of these statistics versus time (e.g., utilization each hour) proves invaluable in two ways. First, it aids in debugging the model since occurrences such as an AGV stopping operation unexpectedly during a simulation run are highlighted. Secondly, trend data is necessary for thorough system analysis. It can readily show the maximum number of moves made in the pre-set time increment (important data for peak throughput capability determination) or the maximum interference incurred. A review of the AGV charging trend indicates whether an AGV is overutilized for periods of time shorter than the total simulation run. For example, a run may show that the AGV was opportunity charging 70% of the time (on average), but for a period of 10 hours it never had a chance to charge. This extended period of high utilization may be impractical, but it is not apparent in the standard statistical output.

An example of trend output showing hourly (the upper plot line) and cumulative AGV charging time is presented in Figure 8. Note there is a point of total AGV battery discharge at day 21.

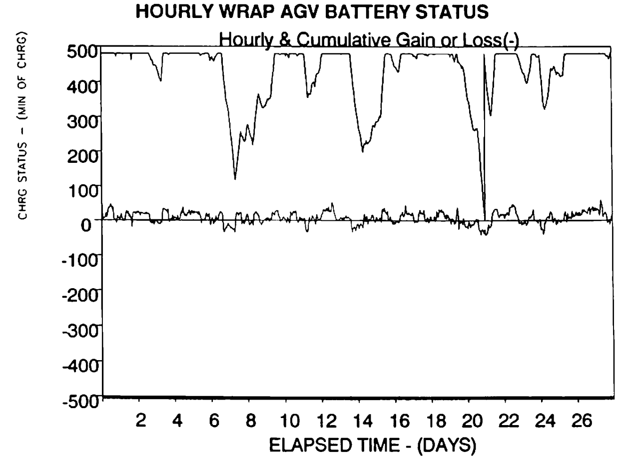

Figure 8. AGV Charging Trend Output

10. ASRS OUTPUT STATISTICS

The performance of the ASRS is measured in two areas – the individual crane performance and the ASRS inventories. The number of moves are tabulated for each ASRS crane; this might be further broken down by type of load, and stores versus retrievals. (The difference between the latter two equals the net inventory change over the duration of the simulation run). A further measure of efficiency in an ASRS system is the number of double moves a crane makes. For instance it is usually desirable for a crane about to store a load, to determine if it has a load to retrieve, and if so, store its load as close as possible to the retrieve load, and then proceed to make the retrieval even if some other store request preceded the retrieval request. Capturing these double moves provides a measure of system efficiency under many different facility operating modes.

It is necessary to tabulate utilization and interference as time-persistent statistics. The inventory level is a primary concern in ASRS systems. This is a continuous change time-persistent variable, and average, maximum, minimum, and variation data are captured. The statistics are tabulated for all appropriate subsets as well as the total inventory.

Trend data is analyzed for each of the ASRS system performance measures. A plot of inventory level is particularly useful since it readily shows any undesirable long term build-up or depletion trend. The standard statistical output defines the maximum level, but not when it occurred, or how often, or if there is a constantly increasing or decreasing trend.

11. OBTAINING TREND OUTPUT

As previously mentioned, simulation software packages provide a mechanism to obtain standard output which will satisfactorily provide counts, and discrete and continuous variable statistics. However, obtaining periodic observations to generate a plot of the performance measure versus time often leaves the modeler to his own devices, particularly if there are more variables then the number of open files allowed by the operating system being utilized.

In such a case, code must be written to output the observations to one or more files during the simulation run, and then post-run manipulation is required to create the desired plots. For example, when using SIMAN, an entity is created which calls a FORTRAN subroutine to open an output file and schedule the first call of a second subroutine which actually calculates the statistics every predefined period of simulated time (e.g., an hour). The FORTRAN subroutine then uses the SIMAN system variables (e.g., NC(#), NQ(#), TAVG(#), DAVG(#), etc.) to calculate the value for the previous hour. The output file(s) is an ASCII file(s) which is easily imported into spreadsheet software where plots can be constructed and printed.

12. CONCLUSIONS

Design and development of an AGV and ASRS simulation that provides for the input and output characteristics outlined above results 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.

Developing a model which provides for these input and output factors pays dividends, particularly when flexibility is an objective early in the design. Significant savings in time and model size are obtained. A continual effort to improve simulation efficiency, while maintaining goals of meeting client expectations is especially effective when modeling is started early in a facility's design and engineering phase when the only constant in the project is change itself.

APPENDIX A. REGULATING AGV INTERFERENCE

The logic shown in Figure A.1 regulates interferences between AGVs and other transporters and is used sucessfully in our simulation software. The decision logic is a stand-alone module not requiring adaptation to any track or circumstantial events that we have yet encountered.

The routine requires prior knowledge of the subject AGV number and indexed structure of AGV routes, their component segments, and identification of interference sets that group segments in order to prohibit unauthorized access by another transporter.

The routine assumes that a subject AGV needs access to a segment which an oncoming AGV has claimed. The actual code contains a loop that indexes through the oncoming AGV's unfinished route to determine its time of arrival to that segment. The subject AGV's time to this point has already computed (the subject AGV need not be on this segment, it could be checking the segment's anticipated availability).

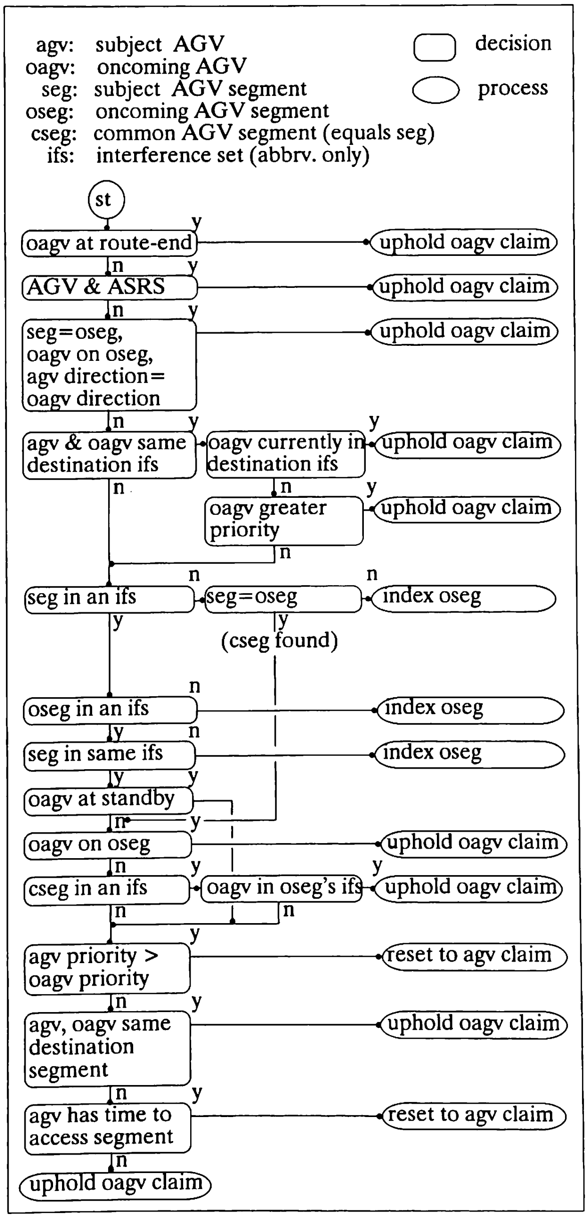

Figure A.1 Logic Flow

APPENDIX B. STRUCTURE AND OPERATION OF ASRS INVENTORY ARRAY

The heart of the inventory system is the structure of the file that contains bin information. The structure (see Figure B.1) consists of data fields serving two purposes: a sequential listing of bins and their contents (column 1); and tables listing bin contents, but sorted into product types (columns 2-5 each represent a different product). In this way, information of specific bin numbers can be accessed as well as sub-inventories of every product type.

bin #	1	2	3	4	5
7	0 n/a			17	
6	-30004 e			149	
5	30002 cf	-5		303	
4	-30277 e	6	112	255	
3	30000 f	68	92	129	
2	30000 f	137	128	66	122
1	-30000 ce	12	3	98	44
	Bins (a)	Empty Bins (b)	Prod1 Inven (c)	Prod2 Inven (c)	Prod3 Inven (c)

array column #

Legend

(a) e: empty (-30000 - prodTYPErecord#)
f: full (30000)
ce: claimed empty (-30000)
cf: claimed full (30000+ prodTYPErecord#)
n/a: unused (0)

(b) +: pointer to bin#
- : pointer to claimed full bin only
Note : claimed empty bins do not need a pointer to bin#

(c) +: pointer to bin#

Figure B.1 ASRS Inventory Array

To illustrate the system, assume an item to be stored; thus, a bin# is needed. The system first accesses the column2 listing of empty bins, (empty bins are considered a product type) and randomly picks 4 (a number between 1 and 5, the total empty bins; it is re-chosen if the one containing -5 is picked, denoting a claimed bin). Bin 4 in column 2 contains the pointer 6, referring to bin 6 in column 1. Bin 6 is then set to claimed status, and the empty bin in column 2 is overwritten. Note that bin 6 also points back to its spot in the empty bin listing. This dual cross-referencing is necessary during crane double-moves when the desired bin# is already known and not picked randomly. The array is created and initialized at simulation startup.

REFERENCES

Pegden, C.D. (1986), *Introduction to SIMAN,* Systems Modeling Corporation, State College, PA.

Sadowski, R.P. (1989), "The Simulation Process: Avoiding the Problems and Pitfalls," In *Proceedings of the 1989 Winter Simulation Conference,* E.A. MacNair, K.J. Musselman, and P.Heidelberger, Eds. IEEE, Piscataway, NJ, 72-79.

Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

A CASE STUDY: SIMULATION OF PACKAGING LINE CONTROL LOGIC

Stuart Gittlitz

Kraft General Foods
General Foods USA Division
250 North Street
White Plains, New York 10625

ABSTRACT

Kraft General Foods has been utilizing simulation techniques for the past several years. In our simulation of high speed packaging lines, one area that has provided efficiency improvements is the evaluation of control logic. This paper presents a case study in the form of a mystery. After a discussion of the packaging line's designed strategy and control logic, we will use simulation to illustrate why this logic concept did not actually work. The line operators had disabled portions of the disfunctional logic and were controlling the line in an altered sequence from the original design. The paper discusses the reasons for the operators' actions as well as an equitable solution.

1. INTRODUCTION

Since 1986, General Foods (now a division of Kraft General Foods) has been developing the use of simulation in evaluating and improving the efficiency of high speed packaging lines. Off-the-shelf simulation packages have not provided the accuracy required to analyze the complex detail of our lines. We have developed a generic simulation shell with Pritsker Corporation that is currently being used in several of our plants. This paper addresses the use of simulation to evaluate and improve the control logic of the line.

How do we define control logic? It is made up of two distinct parts, human logic and programmed logic. The human logic would include a multitude of management and operator decisions: when to take a troubled machine out of service for repair, how to adjust Filler speeds to insure that label weights are met, and the start-up sequencing of multiple lines. Programmed logic refers to the use of photo-eyes and programmable controls as part of the packaging line's operating system. The photo-eyes are positioned on the line to react when product "backs up" in front of the eye (the eye is covered) or "clears" the area in front of the eye (the eye is uncovered). Programmed logic is set into action when an eye is covered or uncovered for a specified length of time (a delay before taking action). The logic could start or stop machines, change speeds, activate surge systems, stop conveyors, or respond in a variety of other line functions.

This paper concentrates on the simulation of programmed logic. The case study included here has been disguised and simplified for proprietary reasons, but the essence of the study is intact. As packaging lines have been modernized with programmed logic, there have been opportunities to reduce crewing and to standardize logic decisions. A main drawback has been the difficulty of an observer to evaluate errors in the design or application of the control logic. Changes to the line may have caused the logic to be obsolete for the new conditions. The strategy of the logic design may not be known to the line operators or management. The programmer responsible for maintenance of the line controls talks in a language of "ladder logic". We have used simulation to gain an understanding of the programmed logic for discussion with both the programmer and line management. The simulation allows faults in the logic to become readily observable and permits an experimentation procedure to improve the logic.

The case study reviews a packaging line used to fill bottles. The simplified version of the line includes only three machines, two Surge Tables, and eight Photo-eyes. A discussion of the strategy of even this simplified line becomes quite complex. However, an understanding of the line is required to evaluate why the production operators disabled the logic for actual factory floor conditions. We will examine both the design and actual strategy in detail. Several solutions will be discussed and presented.

2. DESCRIPTION OF THE "BOTTLE LINE"

The bottling line in Figure 1 fills bottles, combines the bottles in a 3-pack package, applies a label, and transfers the 3-pack to downstream operations (case packing, palletizing, etc.) There are two surge tables on the line: 1) between the Filler and Combiner and 2) between the Combiner and Labeler. When the conveyor between the Combiner and Filler becomes

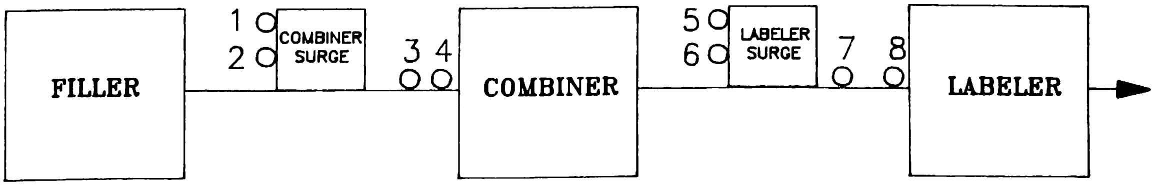

Figure 1. The "Bottle Line"; Photo-eye Locations are Represented by Circles