Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

GENERALIZATION OF AN AS/RS MODEL IN SIMAN/CINEMA

Ali Gunal
Eric S. Grajo
David Blanck

Production Modeling Corporation
One Parklane Blvd. 1604 W
Dearborn, Michigan 48126, U.S.A.

ABSTRACT

A model of an Automated Storage and Retrieval System built using the SIMAN/CINEMA simulation software is presented in this paper. A detailed description of the system is given. The modeling issues that are raised by the nature of the system and by the requirements of the analyses to be made are explained. As the model involves both SIMAN and FORTRAN code, a description of various tasks accomplished by both parts is provided. Furthermore, a brief discussion of some of the results obtained by using the model is presented. Conclusions of the study regarding a generic approach to modeling similar systems are given at the end of the paper.

Key Words: Facilities Planning, Materials Handling, Warehousing and Distribution

1 INTRODUCTION

The objective of this paper is to describe an approach to build a generalized model in SIMAN/CINEMA that can be used in simulating different Automated Storage and Retrieval Systems (AS/RS) with minimal or no modification to the code. The method introduced in the paper has evolved during a simulation study of a large scale AS/RS and it consists of a SIMAN model supported by a set of FORTRAN routines. The subject AS/RS is designed to act as a buffer storage between the two major phases of a manufacturing process. In this paper, the system and the model are described in as much detail as possible with the objective of demonstrating the approach.

There are reports of several successful applications of simulation in studying similar AS/RS, e.g. Mederios, Enscore, and Smith (1986), Muller (1989) and Harmonosky and Sadowski (1984). Ragunath, Perry, and Cullinane (1986) present a flexible approach for modeling a class of AS/R. The approach taken in this study is very similar in nature to the detailed modeling approach described by Muller (1989) in his discussion of various approaches to modeling those systems at different levels of detail.

In the next section, the subject AS/RS is described along with the flow of materials in and out of it. Then, in section 3, the modeling issues that arose as a result of the nature of the system as well as analysis requirements are discussed. The model with its SIMAN and FORTRAN components is described in section 4 with a discussion of how the issues mentioned in section 3 are addressed. Some of the results obtained using the model are briefly explained in section 5. The features of the model that make it applicable to similar systems are discussed in Section 6. Finally, the paper ends with a discussion of some of the important conclusions of the study.

2 SYSTEM

2.1 The Storage System and the Conveyor Interface

The AS/R system is composed of a seven aisle rack system with 6 tiers in each rack and 15 bays in each tier. Since storage and retrievals are possible to/from the racks on either side of an aisle, each aisle consists of a total of 180 bays with 90 on each side. The storage system is linked to the rest of the system through a peripheral conveyor system. Each aisle of the system is interfaced with the peripheral conveyor system through pick and drop stations (P&D) at both ends of the aisle.

Each aisle of the system is served by a crane which can move in horizontal and vertical directions simultaneously. Once the crane is positioned at a desired location for storage/retrieval to/from a certain bay, the total time to extend its shuttle into the bay, to drop/pick the part and to pull the shuttle back is 11 seconds (called shuttle cycle). The same timing applies to picking/dropping a part from/to the pick and drop station. The parts flow in and out of the system at two levels: at the ground level (same level as the first tier of racks), and at the second level which is at the same height as the third tier of the racks. A sketch of the system layout at each of these levels is provided in Figures 1 and 2. As a convention, the aisles are numbered 1 through 7 starting from the top. The bays at each level of the rack system are numbered 1 through 15 starting from the east end.

Every inbound part is first moved onto a pick and drop station at either end of an aisle and then picked by the crane and moved to its destination bay. An outbound part is picked from its bay in the rack system by the crane and is transferred to the pick and drop station at either end of the aisle. Once dropped on a P&D station, the part is moved onto the outbound conveyor by a small roller conveyor paired with a lift table that can lift and lower parts from/to the conveyor. There are also hold tables to serve as temporary stop points along each peripheral conveyor. There is a set of two conveyors at both levels of the east end of the system. The conveyor on the left runs from north to south, and the other from south to north. There is only one conveyor at each level of the west end of the system. Both of these conveyors run from north to south.

2.2 Material Flow

As a convention, the part types flowing in and out of the system are classified into four as types A, B, C and D. A further classification of type A parts is made as A1, A2, and A3, and of type B parts as B1, B2, and B3. The system is designed to operate through two shifts of operation and the flow of materials follow different patterns in each shift. A detailed description of the flow of parts during both shifts is given below.

As mentioned earlier, the system is between two major phases of the manufacturing process. These steps are labeled as P1 and P2. A further classification based on part types is given as follows: Following the completion of process P1A, all type A parts arrive to the system at the north end of the southbound conveyor at the west side of the second level. The destination aisle is assigned on a round-robin basis starting from the first aisle (i.e., first part goes to first aisle, second part goes to second aisle, ..., eighth part goes to first aisle, and so on). The retrievals of type A1 and type A2 parts are triggered from process P2a12 whereas retrievals of type A3 parts are requested by P2a3. All type A parts leaving the system are moved to the southbound conveyor at the first level of the west end of the system. Once placed on the conveyor, all of these parts travel to south, and then type A1 and A2 parts proceed west for process P2a12 and type A3 parts proceed east for process P2a3. The flow of type A parts in and out of the system follows the same pattern and rate during both shifts.

An important difference between type A and type B parts is that a portion of type B parts have to visit process P1B more than once before they are ready for process P2B. Furthermore, the operation at process P1B runs through both shifts but process P2B runs through the first shift only. Consequently, during second shift the system builds up a stock of type B parts.

Type B parts, which are further classified as B1, B2, and B3, arrive to the system from process P1B at the north side of the system.Then they travel to west finally reaching the north corner of the inbound conveyor at the second level of the east end of the system. The destination aisle is assigned on a round robin basis. The retrievals of type B parts which completed their process in step P1B are triggered by process P2B. Those parts leave the system at the first level of the east end. Type B1 parts travel to the south end of the southbound conveyor and then they cross over the northbound conveyor and proceed to east to process P2b1. Type B2 parts stay on the southbound conveyor until they reach to the sixth (counting from the top) lift table on the conveyor, where they are moved east towards process P2b2 crossing over the northbound conveyor. Type B3 parts are moved directly to the northbound conveyor crossing over the southbound conveyor once they leave the pick and drop station at the east end of an aisle. Then they travel to the end of the northbound conveyor where they are routed to process P2b3. The retrievals of those type B parts that need to revisit process P1B are triggered by the process itself. Those parts are moved onto the northbound conveyor at the second level by crossing over the southbound conveyor. Once they reach to the end of the northbound conveyor, they proceed west to process P1B.

Type C parts arrive to the system from east of the north corner of the southbound conveyor at the second level during the first shift only. Clearly, at the beginning of the southbound conveyor, type C parts are mixed with type B parts upon arriving to the system. An aisle assignment on a round robin basis is made only after a part is moved to the lift table at the north end of the southbound conveyor. The retrievals of type C parts are made only during the second shift. Hence, there is a build up of stock for type C parts during the first shift. During the second shift these parts are removed from the system and placed on the northbound conveyor at the second level. Then, they leave northbound conveyor at the north end and proceed east.

The inbound parts of type D follow the same route as type C parts. Similarly, these parts arrive in the system during the first shift and leave the system during

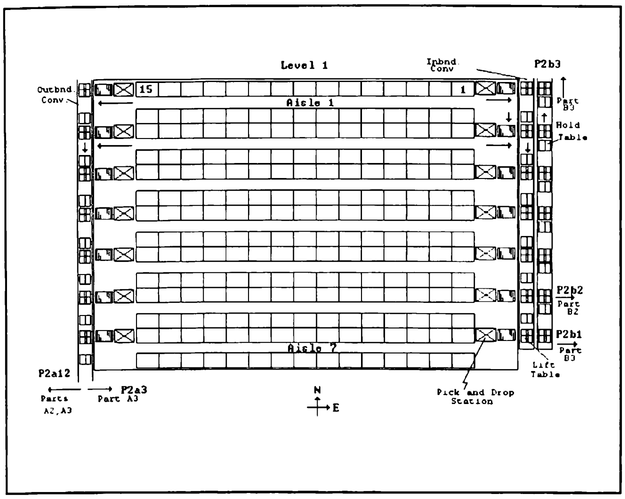

Figure 1: The layout and flow of materials at the first level of AS/RS.

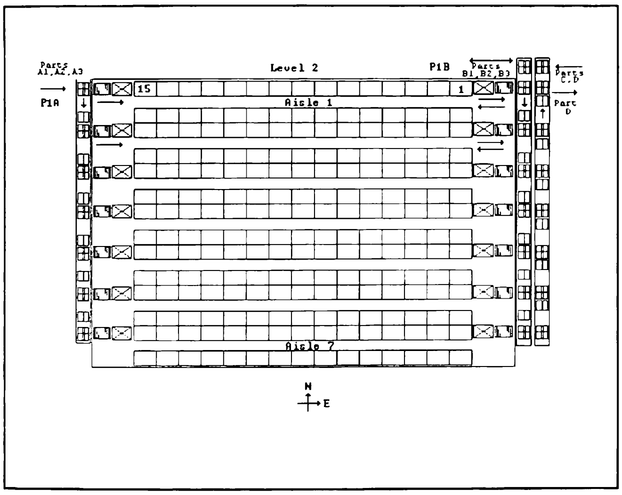

Figure 2: The layout and flow of materials at the second level of AS/RS.

the second shift. Type D parts leaving the system follow the same path as outbound type C parts with the exception that they leave the northbound conveyor at the lift table before the last one, and proceed east.

3 MODELING ISSUES

A detailed examination of the system requirements reveals several interesting modeling issues that should be addressed for an accurate representation of the system. The following is a description of those issues that are addressed in the present model of the system. The way those issues are tackled are explained in the next section.

1. Most of the parts to be stored in the system have similar physical dimensions except for type A2 parts which are significantly bigger than the others. To accommodate those parts the racks are designed so that bays 13 through 15 (those at the west end) of each tier of the racks are larger than bays 1 through 12. Obviously, this means that type A2 parts can only be stored in one of the bays 13 through 15. Although small parts can be stored in any of those bays, it was decided that they (anything other than type A2) would be stored only in small bays unless there is an emergency. All of these bays are at the west end of the rack system obviously affecting the storage and retrieval times in favor of A2 parts against A1 and A3 parts.

2. The control logic of the AS/RS system is designed so that a storage is made to the bay closest in travel time to the P&D station where the arriving part is picked up. This requirement imposes two related constraints on the simulation model. First of all, the model needs to know the time it takes to travel to a given bay from a given pick and drop station or in general the time it takes to travel from a given position in the aisle to another. Secondly, it should perform a search of the available bays to determine the open location closest to a given P&D station. Although the travel times can be approximated by using expected value formulas similar to those presented by Bozer and White (1984), a closer representation of the behavior of the system can be made by doing those calculations during the simulation run based on the distances to be traveled and the acceleration and deceleration of the crane. The AS/RS in question is required to make more than 4000 storage and retrievals in each of the two shifts it operates. Thus, finding an efficient way of making those calculations is a necessity for a simulation model that is of some practical value in making timely analysis of the system under varying conditions.

A related problem is the requirement that the model reports as part of its output the percentage of dual crane cycles in each aisle. A dual crane cycle is defined to be either a retrieval immediately followed by a storage to/from the same P&D station or a storage immediately followed by a retrieval to/from the same bay or a bay in close vicinity. Such a calculation obviously requires the model to track the position of each crane in each aisle as well as the nature of the last crane move.

3. The process P2 requires that the parts reach to the beginning of the process in the sequence that they arrived in the AS/RS. Since parts can not pass each other on the conveyor system they should be released to the outbound conveyor in a manner that will satisfy this requirement. Once a part is ready to move onto the outbound

conveyor, there may be two cases regarding the position of the immediately preceding part: i) it may be coming out of an upstream aisle, or ii) it may be coming from a downstream aisle. In the first case, the current part should wait until the preceding part passes by the aisle of the current part. In terms of model requirements, this constraint translates into a mechanism that will block the current part from moving onto the conveyor until the preceding part passes by, and then to move the part onto the outbound conveyor as soon as its predecessor passes by. In the second case, there should be a mechanism to activate the movement of the current part as soon as the preceding part starts moving onto the outbound conveyor from the downstream aisle. It is noted that the situation is slightly complicated at the east side since outbound parts move in both directions (e.g., type B1 moves toward south, and type B3 travels to north) whereas at the west side the movement of outbound parts is in one direction.

4. Two other issues are due to the requirements that i) the model is to report the utilization of the large and small bays of each aisle, and ii) the model is to have the capability to initialize the system with a given inventory configuration. These limitations combined with the fact that parts are to be retrieved in the order they arrive in the system makes it necessary that the model track the contents of each bay in the system during the entire simulation length.

5. As mentioned earlier, a certain percentage of type B parts need to revisit process P1B more than once. Unlike the retrievals for other destinations which demand one retrieval at a time, the process requires that a batch of type B retrievals be made at once (the batch size varies between 1 and 7 with a given frequency distribution) and that these parts reach P1B as a group.

4 THE MODEL

The AS/RS system described above was modeled using the SIMAN/CINEMA IV simulation software. The movement system around the racks was modeled completely in SIMAN. However, FORTRAN routines were also used to accomplish certain tasks that could be tedious to code using SIMAN language constructs only. A description of the FORTRAN subroutines and the SIMAN model is given below.

4.1 FORTRAN Subroutines

The FORTRAN routines are mainly used to address the problems of i) initializing and tracking the contents of the racks, ii) tracking the position of each crane in each aisle, iii) calculating the travel time for the crane movement between two locations within an aisle (the travel time between any two points is the maximum of the travel time in horizontal and vertical directions), iv) determining where an incoming part should be stored, and v) determining the location for the part that should be retrieved when there is a request. To perform these tasks, several variables and FORTRAN arrays are used. The most important of those are described below.

InBay(7,180): Keeps track of the contents of each bay in the rack system. Each part type is assigned a unique integer number and once a part is in a bay, the corresponding element of the array is set to the integer number that represents the part. This array is initialized by reading from an external file. A value of zero in this array corresponds to an empty bay.

IBayDedi(7, 180): Each element of this array indicates whether the corresponding bay is dedicated for a given part type or not. More specifically, it is used to distinguish small bays from large bays. This array is initialized from an external file.

TimeIn(7, 180): Each element of this array stores the time that the part in the corresponding bay arrived to the system. This array is initialized by randomly generated numbers between 0 and 1 in order to ensure a random retrieval sequence at the beginning of the simulation.

XofBay(15): Each element of this array represents the x-axis position of each bay at every tier. Since the tier height is the same over all 6 tiers, there is no need to have a separate array to store the y-axis coordinates. This array is initialized by reading from an external file.

Closest(4, 180): For each of the four pick and drop stations, this array stores the list of bays in ascending order of their distance in travel time to the station.

XofPD(4): This array is used to store the x-axis coordinate of each of the four pick and drop stations. The first two elements are used to represent the location of the two first-level pick and drop stations whereas the other two are for the second-level stations. This array also is initialized from an external file.

XofCrane(7): Used to track the x-axis location of every crane (initially at location (0,0)).

YofCrane(7): Used to track the y-axis location of every crane.

LastCrM(7): Used to store the type of previous crane move (e.g., either retrieval or storage).

The FORTRAN subroutines utilized by the mode are user subroutines that can be called from a SIMAN model as well as the PRIME and WRAPUP subroutines that are called by the SIMAN simulation engine. Also some utility functions are used by those subroutines. The following is a description of the tasks accomplished by those subroutines and functions.

Subroutine PRIME

This subroutine:

1. Opens and reads data files to initialize arrays InBay, IBayDedi, XofBay, XofPD.
2. Initializes the TimeIn array with random numbers between 0 and 1.
3. For each pick and drop station:
   i. calculates the travel time between the station and each bay in an aisle,
   ii. sorts the bays in ascending order of their closeness to the pick and drop station and stores this order in array Closest.
4. Determines how many small and large bays are full initially and initializes corresponding SIMAN Resources.
5. Initializes other variables and arrays.

Obviously, calculating and sorting the travel times between all pick and drop stations and every bay in the rack system only once at the beginning of the simulation eliminates the need to make similar calculations every time a storage is to take place. Also, by reading most data from external files, the need to change and recompile the FORTRAN routines is minimized.

User Functions 1 Through 7

The SIMAN language allows calls to be made to user-written FORTRAN subroutines through references made to UF(n) (where n is the number of the function) in a SIMAN program. Each call to UF(n) returns a single value unless provisions are made for passing of more than a single value from the FORTRAN routine to the SIMAN model and vice versa. This task can be done through either global SIMAN variables or entity attributes. The present model makes use of entity attributes as a means of such data exchange between the SIMAN code and the FORTRAN subroutines for reasons that will become clear with the discussion of the user-defined functions of the model.

UF(1): This function is called after an inbound part reaches a pick and drop station at any aisle. The function is coded so that it can be used for any aisle and any pick and drop station. Consequently, it requires that the part informs the function regarding the aisle and the pick and drop station. It searches the racks starting from the closest bay to find an open location and then sets an attribute of the part to point to that location. The SIMAN model logic requires that the aisle assignment is made long before the part reaches to the pick and drop station. Hence, once the aisle assignment is made, it becomes an attribute of the part and makes it possible to pass that information to the FORTRAN subroutine without the use of global variables. While searching for an open location, the function also considers whether the current part needs a large bay or not. Once the closest open bay is located, the function sets the status of the bay to be occupied. Since the part is not yet in the rack system, the function also ensures, by setting appropriate flags, that a retrieval from the bay will not be done until the part is physically in the bay. The pointer to the bay is the return value of the function.

UF(2): This function is called after the crane in an aisle is assigned to make a storage from a pick and drop station. The function accepts as an argument the address of the pick and drop station that the part arrived. Then, the function calculates the time it will take the crane to move from its current position in the aisle to the pick and drop station and returns this value to the SIMAN model. The function also updates the appropriate statistics if the current crane move is resulting in a dual crane cycle.

UF(3): The return value of this function is the time that it will take the crane to move an inbound part from the pick and drop station to its assigned bay in the racks. It is called only after the crane arrives to the pick and drop station and picks an inbound part. The arriving part passes to the function the pointers of the present pick and drop station and its destination bay.

UF(4): This function is called after the crane stores an inbound part in its destination bay. It updates the InBay array to reflect that the part is now in the racks and available for retrieval. It also updates the corresponding element of the TimeIn array by setting it to the arrival time of the part to the system.

UF(5): This function is called when there is a retrieval request for a certain type of part. It searches the rack system for the part that arrived earliest. Once such a part is found, its location, in terms of the aisle and the bay, is passed to the SIMAN code in entity attributes. The return value of the function indicates whether a part is found or not. If the search fails, the SIMAN code updates statistics and attempts another retrieval in the next cycle. If a part is found, the function also updates the status of the bay to reflect the fact that the part stored at that location is marked for retrieval. This step is necessary to prevent the reallocation of the bay prematurely as well as designation of the same part for another retrieval request.

UF(6): This function calculates the travel time for the crane from its current position in the aisle to a designated bay. It also updates statistics if the move is resulting in a dual crane cycle. As input, it receives the pointer to a bay in the racks. It is called only after the crane in the aisle is assigned to the retrieval of a part.

UF(7): This function calculates the time the crane will take to move from a bay to a designated pick and drop station. The calling part passes to it pointers to the origin bay and the destination pick and drop stations. It is called after the crane picks the part from its origin bay. The function is also responsible for updating the status of the current bay as available for storage.

Although it may seem that the functions that are used for travel time calculations could be combined into one, the fact that different statistics are to be updated during retrieval and storage cycles requires that a distinction be made. Since it results in simplified logic, the present model is designed to work with separate functions for each of the crane cycles. An obvious alternative is to use additional flag variables and entity attributes to avoid redundancy in code. However, this approach would require several more "if-then-else" statements that would make the code more difficult to understand, debug, verify, or modify subsequently.

Subroutine WRAPUP

This standard SIMAN function was used to perform the following bookkeeping tasks at the end of the simulation run:

1. calculate and report the percentage of dual crane cycles
2. save the contents of the InBay array in a file so that the status of the system at the end of simulation can be examined.

In addition to those functions several other functions were developed to perform such tasks as i) calculating the time it takes to move between any two points in an aisle by using horizontal and vertical acceleration and maximum speed values, ii) converting pointers to bays and pick and drop stations to points in the assumed coordinate system for the racks.

4.2 The SIMAN Model

As mentioned earlier, the movement system (i.e., the peripheral conveyor system and cranes), was modeled using the SIMAN language constructs. In addition to the movement system, the SIMAN code was also used for generating retrieval and storage requests, collecting various performance statistics, and monitoring the status of the system on an hourly basis.

Retrieval and storage requests are generated in the model based on the cycle time and product mix data which shows variation between the first and second shifts. Since distributions can be parameterized in SIMAN (using the ED(n), where n is the number of the distribution as defined in the experimental frame), a single generation loop is sufficient for each part type. A timer segment is used to modify the value of the parameter representing the distribution at the end of each shift so that the correct distribution could be used for generating retrieval and storage requests. One exception to the above scheme is the generation of batch retrievals that are required to return to process P1B. For those parts, the retrieval generation loop continues to generate but holds the requests in a pool until a randomly chosen batch size is reached. When the batch size is reached, all accumulated requests are released at once. Since all the parts in the batch are numbered in succession and since deliveries are made in the retrieval sequence, the model ensures that the batch arrives at the destination point as a group without getting mixed with other retrievals going to the same destination.

In modeling the movement system, the macro station feature of the SIMAN language (Pegden, Shannon, and Sadowski 1990) is found to be extremely useful by considering the fact that a similar logic is involved at every pick and drop station, at every hold table, and at every lift table along a peripheral conveyor segment. One macro station is found to be sufficient to represent the logic at all seven pick and drop stations at either end of the system at either level. Similarly, one macro station is sufficient to represent the control logic at each one of the seven lift tables along one of the conveyor segments. Consequently, it can be argued that the macro station constructs help reduce the code approximately by a factor of seven for the present model.

One of the problems in modeling the movement system is due to the sequenced delivery constraint as explained before (item 3 in "Modeling Issues" section). To effectively model the control logic, a set of WAIT and SIGNAL statements is used in the model in conjunction with a set of status variables. The logic used is explained briefly as follows: Once a part is ready to move onto the outbound conveyor it checks a status variable indicating whether the immediately preceding part has passed by the aisle the current part is coming from. If the result of the check is positive, then the part may proceed without having to wait further. If the result of the initial check is negative then the current part waits until it receives a message from the immediately preceding part indicating that it can proceed. Once it starts moving, the current part sends signals to upstream aisles so that any other part waiting for the current part to move can proceed. Also, the current part, along its way on the outbound conveyor, updates the status indicators at each aisle it passes by and sends signals to any other part waiting for the current part to pass by. It is clear that a separate set of signals should be used for each part type to ensure that the signals intended for one part type do not activate the movement of parts of other types. This requirement is easily satisfied in the model by simply numbering the parts of each type starting from a different base number (i.e., parts of type A1 are numbered from 1 to 9,999, parts of type A2 are numbered from 10,000 to 19,999 etc.) and then generating the signals based on the number of the part.

5 SYSTEM ANALYSIS

The model described above was extensively used to evaluate several alternative configurations of the system which consisted of different numbers of hold and lift tables, alternative routings for different part types, different numbers of pick and drop stations at each aisle, and different levels of initial stock in the system among other parameters. An interesting part of the study was the evaluation of the number of pick and drop stations in each aisle at the east end of the second level of the system. Since the traffic at the east end of the system at second level is bidirectional, a configuration with two pick and drop stations (as shown in Figure 3) instead of one was to be evaluated. Each of the pick and drop stations would be dedicated to one direction of traffic (i.e., either for inbound or outbound parts).

The number of storage and retrievals per aisle was taken to be the primary performance measure along with the number of pending storage/retrieval requests. The nature of the manufacturing process required that the number of pending requests not exceed certain limits on average as well as at maximum. Thus, it was necessary that the standard deviations and maxima as well as the average measures, values were taken into consideration.

The performance measures were collected for 20 shifts of simulated time following a warm-up period of 4 shifts of operation (thus, statistics were collected for more than 20,000 storage and as many retrievals).

A comparison of the confidence intervals constructed by using the batch means method (Law and Kelton 1991) showed a significant difference in performance between the two configurations. Although the averages were close to each other, the confidence intervals constructed for the configuration with one pick and drop station were much larger than those for two pick and drop stations. Also, the maximum number of pending retrieval/storage requests was higher (and beyond acceptable maximum) for one pick and drop station. Consequently, it was concluded that a configuration with two pick and drop stations was a more favorable configuration than the one with one station under the conditions simulated by the model.

6 GENERALIZATION OF THE MODEL

During the course of the study it became apparent that the modeling approach taken in this study could easily be put in a generic form so that similar systems could be modeled by reusing most of the existing code. Particularly, the FORTRAN routines are designed to have general applicability. More specifically, the functions are written so that most system parameters are represented as variables in code. For example, the number of aisles, the number of bays in each tier, the number of tiers, and the number of pick and drop stations can be modified with no change to the FORTRAN code. Although this approach requires that all those parameters be initialized by reading from external files, in most cases it is preferable to changing, recompiling, and relinking the FORTRAN code. A similar approach is taken in constructing the SIMAN part of the model through macro stations and indirect addressing capability of the SIMAN language. These features of SIMAN permit use of single set of statements to represent the flow of entities and related logic in more than one part of the model. Consequently, once developed, a macro station can be used to simulate as many similar stations as needed. In addition, as can be observed in Figures 1 through 3, the system shows a high degree of symmetry from one aisle to another as well as from one conveyor to another. By capitalizing on this observation and on the capabilities of SIMAN, the model is designed so that the number of aisles, the number of tiers, and the number of bays could be changed with only slight modifications to the SIMAN code. For example, part of the statements that describe the logic at each one of the pick and drop stations at the west side of the second level looks like the following:

```
STATION,WMPandD1-WMPandD7;
;-----------------------
ASSIGN: Aisle = M - WMPandD1 + 1;
QUEUE, WMCraneQ1+Aisle-1;
SEIZE: Crane(Aisle);
```

In this segment, the `ASSIGN` statement determines at

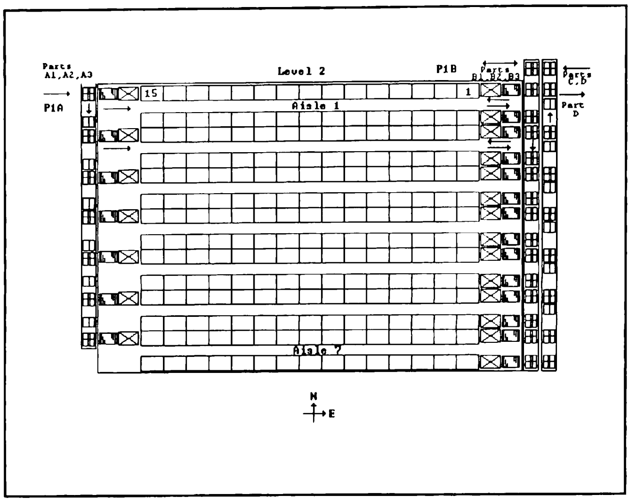

Figure 3: The layout and flow of materials at the second level of AS/RS with 2 pick and drop stations at the east side mezzanine level.

which aisle the part has arrived and stores that value in entity attribute `Aisle`. The second and third statements, by making use of indirect addressing and an indexed resource, ensure that the part captures the crane in that aisle before proceeding further. It is clear that the same code could be used for any number of pick and drop stations other than seven, as presently in the model, with a slight change in the code (by modifying the `STATION` statement, and by inserting/deleting stations in the `STATIONS` element of the experimental frame).

A similar approach is used in achieving a flexible representation of the material flow in and out of the system. This flexibility is accomplished by benefiting from the observations that the material flow pattern is based on part types and that each segment of the conveyor system is unidirectional. Thus, the decision logic at each hold/lift table involves a simple check of the destination point of the current part type. Consequently, the SIMAN model of the movement system (peripheral conveyors, pick and drop stations, cranes etc.) can be designed so that once a part is put at any point in the movement system, it reaches its destination without requiring a complicated routing logic. Obviously, this approach requires that the destination of the part is kept as an attribute of every part and be assigned as soon as a storage/retrieval entity is created. A sample code segment that demonstrates the modeling approach is as follows:

```
STATION, EMSthBndHT1-EMSthBndHT7;
;------------------------------
ASSIGN: Table = M - EMSthBndHT1 + 1;
BRANCH, 1:
  IF,Destination.eq.EMNrthBndLT1,
        EM100:
ELSE, EM200;
```

where `Table` and `Destination` are entity attributes, and `M` is the standard entity attribute specifying the entity's current station. Since the logic applies to each of the seven hold tables along the southbound conveyor at the second level, the first `ASSIGN` statement obtains the pointer to the current hold table. Then, the `BRANCH` statement checks the destination information to determine where to send the current part.

A second method in developing a generic model is to use a modular structure in writing the code. A high degree of modularity can be achieved by first identifying the primary functions to be utilized in making decisions throughout the simulation model. Once those functions are identified, a separate code segment is developed for each such function even though doing so may result in redundant code. For example, for the subject AS/RS, some of the primary decisions involve questions such as where to store an inbound part (i.e., which aisle, tier, and bay), when and which part to retrieve next (aisle, tier, and bay), and when to release an outbound part to the peripheral conveyor. In addition to isolation of primary functions in code, the use of global variables should be avoided whenever possible to share data between various parts of the program. If and when possible such data exchanges should be limited to passing arguments between the functions.

Since both the SIMAN and FORTRAN parts of the present model are developed around those principles, it would be possible to replace any of them with a different function without disturbing the rest of the system. For example, the algorithm used in the present model to search for a part in response to a retrieval request can easily be replaced with a more complicated one without having to change the rest of the model. Similarly, the algorithm to find an open bay for a storage can be modified to account for more complicated decision rules. Furthermore, it would be possible to use more than one function to perform the same task in different ways. For example, a different decision rule for choosing an open bay based on the types of parts can be easily added to the model. However, care must be taken in designing new modules to replace the existing ones. Special attention must be paid to which variable or entity attributes are used in passing values back and forth between the SIMAN and FORTRAN parts of the model. A clear documentation of entity attributes and system variables utilized in such data exchanges must be kept along with the code so that problems encountered in reusing such code are minimized. It is also important that the input data requirements of the model (both FORTRAN and SIMAN) are documented as well.

7 CONCLUSIONS

The AS/RS modeled in this study is one of the more complicated systems that the authors encountered. During the process of modeling and analysis of the system, several interesting conclusions have been reached. First of all, it became apparent that the modeling approach taken in this study could easily be put in a generic form by utilizing the principles outlined in Section 6. The model was generic in the sense that an AS/RS different only in size from the one described in the paper could be simulated without any modification to the model. In addition, the material flow pattern could be altered by only changing the destination point assignment when a storage/retrieval was created. Furthermore, it was observed that the structure of the existing model provided a strong foundation in modeling similar systems.

The study also showed that the capability to interface with a general programming language was a desirable attribute of a simulation software. The model described in this paper would obviously be less efficient if all the details of the model were coded in SIMAN or in any other simulation language. Also, the use of a general programming language facilitated a highly modular structure in modeling complicated decision logic through the use of subroutine and function calls, which in turn contributed to the flexibility of the model.

Another useful feature of the SIMAN language was its capability to support the "Macro Station" concept. As mentioned earlier, the model described in the paper would be much longer in SIMAN code and it would require significantly more amount of work to make any modifications to the flow of materials in the system.

Although it was time consuming to build, the animation model of the system was found to be very useful in communicating the model logic to the related plant personnel. Furthermore, the animation model proved to be an excellent tool to facilitate fruitful discussions during the design process of the system.

REFERENCES

Averill M. Law and W. David Kelton, 1991, Simulation Modeling and Analysis, Second Edition, McGraw-Hill Inc., New York.

Catherine M. Harmonosky and Randall P. Sadowski, 1984, A Simulation Model and Analysis: Integrating AGV's with Non-Automated Material Handling, Proceedings of the 1984 Winter Simulation Conference.

D. J. Medeiros, E. Emory Enscore, Jr., and Alan Smith, 1986, Performance Analysis of Miniload Systems, Proceedings of the 1986 Winter Simulation Conference.

Daniel Muller, 1989, AS/RS and Warehouse Modeling, Proceedings of the 1989 Winter Simulation Conference.

C. Dennis Pegden, Robert E. Shannon, and Randall P. Sadowski, 1990, Introduction to Simulation Using SIMAN, McGraw-Hill, Inc.

Sethuraman Ragunath, Ronald Perry, and Thomas Cullinane, 1986, Interactive Simulation Modeling of Automated Storage/Retrieval Systems, Proceedings of the 1986 Winter Simulation Conference.

Yavuz Bozer and John A. White, 1984, Travel-Time Models for Automated Storage/Retrieval Systems, IIE Transactions, Vol. 16, No. 4.

AUTHOR BIOGRAPHIES

ALI GUNAL is a Systems Consultant at Production Modeling Corporation. Prior to joining PMC in 1992, Mr. Gunal worked as an Operations Research Consultant for the state of Washington. He received his Ph.D. in Industrial Engineering from Texas Tech University in 1991. Mr. Gunal's interests include robotics and discrete event simulation, manufacturing systems, and scheduling. He is a member of ORSA, TIMS, and SME.

ERIC S. GRAJO is a Systems Consultant at Production Modeling Corporation, an industrial engineering and simulation consulting firm in Dearborn, Michigan. Mr. Grajo specializes in computer simulation and animation models of various manufacturing and material handling applications in the automotive, electronics, food, pharmaceutical, and service industries, including health care. He has expertise in SIMAN/Cinema, Slamsystem, ProModelPC, Automod, QUEST, and SIMFACTORY simulation languages and packages. Prior to joining PMC in 1988, he also had experience in physical distribution and materials management in the packaging products industry. Mr. Grajo has an MSE in Industrial and Systems Engineering and has completed PhD course work requirements in Industrial and Operations Engineering at the University of Michigan.

DAVID BLANCK is an Applications Engineer at Production Modeling Corporation. He specializes in computer simulation models for the automotive, electronics, and health care industries. Mr. Blanck, who joined PMC in 1991, has a BS in Industrial and Systems Engineering from the University of Michigan - Dearborn.