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

AN AGVS SIMULATION CODE GENERATOR FOR MANUFACTURING APPLICATIONS

Dah-Chuan Gong
Leon F. McGinnis

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205

ABSTRACT

Designing an automated guided vehicle application is a complex task. Simulation seems to be the only method of analysis which can give a detailed and accurate prediction of system performance. However, the time spent on modeling and creating a simulation program can be significant. Therefore, the concept of developing an "automatic simulation" method is appealing. This paper describes a simulation code generator (SCG) implemented to demonstrate this concept. 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 uni-directional 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 and to demonstrate its capability.

1. INTRODUCTION

In a manufacturing system, raw materials are converted into finished products by a set of processing steps. Because the processing capability of a typical resource is limited, multiple processing operations are required. So, a transport system is employed to deliver materials between resources (machines). This transport system is expected to provide sufficient performance without limiting the system throughput or causing excessive work-in-process.

For the sake of maintaining flexibility and mobility, the powered carrier is the most popular equipment for inter-resource deliveries. An automated guided vehicle system (AGVS) is one representative of the current technology. An AGV is a driverless, battery-powered vehicle with programmable capabilities for guidance and steering. An AGV accomplishes a delivery by a sequence of asynchronous movements.

However, designing an AGVS is a complex task. It involves more than simply considering flow rates among workstations under a known production plan. The interaction of the AGV system with the production system also must be considered. In designing an AGV application, several questions must be answered. These questions include: (1) How many vehicles are required? (2) What is the guidepath network configuration? (3) What control logic will be used for dispatching vehicles? and (4) Which route should a vehicle follow for a given origin and destination?

Because of the complexity of these problems and their inter-relationships, simulation has become the main method for obtaining a detailed and accurate estimate of performance for a proposed system. Even though simulation is being widely used in system design, planning, and control, the time required for modeling and creating a simulation program can be substantial. It may be viewed as uneconomical, especially for those unfamiliar with simulation and programming practices. Therefore, the concept of developing an "automatic simulation" method is appealing.

1.1 Automatic Simulation

Ideally, an "automatic simulation" operates as a black box. A problem description is the input, and the desired statistical estimates of performance are the output. There are two basic strategies for automatic simulation. One is to create a very general simulation model which is "table driven", i.e., data tables for any problem instance (within the scope of the model) are the simulation input. Note that the generic simulation model (also called simulator) need not be recompiled for each problem instance. The other approach is to develop a generic simulation code generator, or SCG. The SCG, in essence, "compiles" the problem description file into a simulation code, which then is itself compiled into executable code. The SCG itself might be developed using a general purpose language, such as FORTRAN, Pascal, C, or Basic. The target simulation language might be GPSS, SIMAN, SLAM II, SIMSCRIPT, or any other simulation language. Alternatively, it may generate code in a general purpose language.

Both strategies have been the topic of prior research. Gaskins and Tanchoco [1989] developed a C-based discrete-event simulator for AGVS controller design. A FORTRAN-based simulation code generator, DRAFT, was presented by Mathewson [1985]. The DRAFT family consists of a group of modular units for data input/edit, model analysis, and program writing. By different code writer modules. DRAFT has demonstrated its capability for supporting SIMON, GASP II, SIMULA, and SIMSCRIPT. Clementson [1986] introduces the British approach to discrete simulation which is used in conjunction with the CAPS/ECSL package. Models are formulated with an Activity Cycle Diagram (ACD). CAPS, standing for Computer Aided Programming of Simulation, requests the model data in a user friendly conversational mode and then generates a simulation code in ECSL. Balmer [1987] discusses the general concept of an integrated software support environment for simulation modelling. In his computer aided simulation modelling (CASM) group, an interactive simulation program generator (ISPG) is created which also accepts a model specification in terms of an ACD. Haddock [1987] presents an SCG for flexible manufacturing systems (FMS) design, with SIMAN as the target simulation language.

1.2 Overview

This paper also describes the implementation of an SCG to prove the concept of automatic simulation of AGV systems. This SCG is designed in a modular fashion which provides a flexible potential for future expansion. An ASCII database containing a complete problem description is the input of the SCG. The output simulation code from the SCG is in SIMAN [Pegden 1987] syntax. The user is allowed to update the database to reflect the partial change of an existing design. The remainder of this report is organized as follows. Section 2 describes the problem domain. AGVS operational issues are discussed in Section 3. Section 4 addresses the details of SCG implementation. A case study of the SCG is presented in Section 5. The limitation of current SCG version is stated in Section 6 and the future work is presented in Section 7.

2. PROBLEM DOMAIN

In AGVS, a delivery activity begins when a part leaves the buffer (i.e., the storage space located in front of a workstation) of its origin and is loaded onto a vehicle. The completion of a delivery is defined as a part arriving to the buffer of its destination. In the following, the problem domain will be manufacturing system, including the interface (buffer) with AGVS, and the AGVS.

2.1 Manufacturing System Configuration

A shop is the place supplied with resources necessary for production. Production processing involves jobs. A work cell is capable of completing operations of the jobs that have been assigned on it. Inside a work cell, resources (machines) which are not necessary identical are employed to perform operations.

There are three kinds of work cells: storage cells (e.g., AS/RS), fabrication cells, and assembly cells. Only the first two are considered in this paper. A simple work cell contains only one machine. A general configuration for a simple work cell is shown in Figure 1. In front of the machine, there are machine input and output buffers (denoted as BMI and BMO respectively), internal input and output buffers (BVI and BVO), and external input and output buffers (EI and EO). BVI and BVO correspond to the logical representation of the interface between the manufacturing system and the AGVS. BVI is treated as the deposit station where a delivered part is unloaded from a vehicle, and BVO is the pickup station where a part is loaded onto a vehicle.

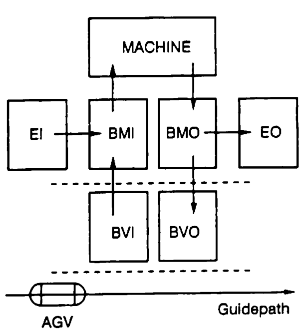

Figure 1. Simple Work Cell Configuration

2.2 Discussion of AGVS

Generally, an AGV system contains seven major components falling in four categories:

(1) the transport network
  (i) the guidepath segments,
  (ii) the work cell pickup/deposit stations,
(2) the vehicles
  (i) the vehicle type, capacity, speed characters, and operation features
(3) the interfaces between the production system and AGVS
  (i) the delivery service request list and dynamic buffer status information,
(4) control system
  (i) the vehicle dispatching control logic,
  (ii) the traffic management, and
  (iii) the navigation.

There are six basic types of automated guided vehicles: unit load, towing, pallet truck, fork truck, light load, and assembly line vehicles [Miller 1987]. With features such as small floor space (narrow aisle) requirement, fast speed, high maneuverability (reduced turn and sweep radii), bi-directional travel capability, and variable load capacities, unit load vehicles have become more popular recently. Therefore, applications with unit load vehicles moving along a uni-directional guidepath network through work cells are the systems of interest.

3. AGVS OPERATIONAL ISSUES

In an AGVS, a work cell is treated as the smallest unit (block) of the production system. Operations on a part may be performed by several machines in a work cell, but the movement of the part within the work cell is not performed by an AGV. The AGV only transports parts in and out of the work cell, and is the only physical connection to the external environment of this work cell. In other words, each work cell can have its own transport device to handle its internal part movements, and this internal movement is independent of other work cells.

Logically, a work cell is represented as a block in which parts are sent out and received via a transport system. As shown in Figure 2, each block (i.e., work cell) may consist of many pickup (P) and deposit (D) stations, or combined stations with the functions as P and D stations. From the material flow viewpoint, these P/D stations are also responsible for connecting (integrating) both the production system (PS) and AGVS. A part coming to a work cell may be delivered to one of the D stations. After a sequence of operations have been performed on different machines in this cell, the outgoing part is moved to a P station and waits for a transporter to deliver it to another work cell.

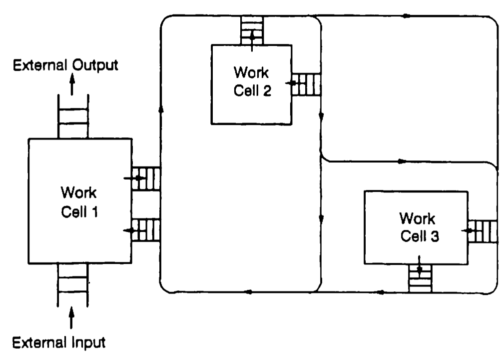

Figure 2. A Shop Configuration in Terms of Work Cell

The transport activities on the shop floor concern deliveries of parts from one work cell to another. A service request is required to invoke a vehicle's movement, and is created by the production system when a part move is required. A service request identification is attached to this move request. Each identification corresponds to a unique part. The associated part information such as the current location and its destination can be uniquely retrieved just by the service request ID.

An AGVS operation scheme which demonstrates the integration with the production system from the information flow viewpoint, is shown in Figure 3. This scheme contains four modules: the production system module (PSM), information module (IM), control module (CM), and the vehicle process module (VPM).

The first module contains information associated with the production system (e.g., part process routes, processing times, and work cell capacities). Creating service requests is one main function of this module. A message of the new request creation has to be transmitted to the information module. At the end of the delivery, a message representing completion of the request is returned from the IM to the PSM.

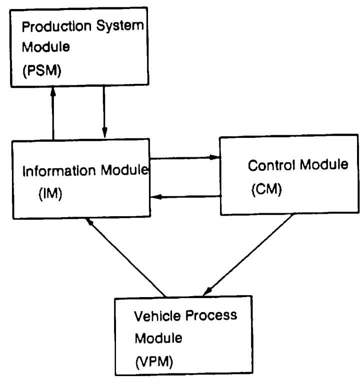

Figure 3. AGVS Operation Scheme

The IM serves as the AGVS data manager. It tracks the system status, monitors the system performance, provides the control module with the essential information for making decisions, and communicates with the production system to exchange messages about the creation or completion of a request.

Once a part move request is transmitted to the information module from the PSM, the IM then adds this request to a list. It also starts to prepare the information required by the control module for planning a vehicle's movement. The control module is responsible for all the decisions related to vehicle assignment, navigation path determination, and the traffic management. It also has to report the results of operations to the IM so the system status can be maintained.

The vehicle process module implements the vehicle movement required by decisions of CM. As a traffic blocking situation occurs during the vehicle's travel, the traffic congestion information is collected and fed back to IM from VPM. This congestion information is then modified in the IM by adding some historical traffic records and is passed to CM. Based on the modified information, the CM may make decisions to release the congestion.

In an AGVS, the vehicle dispatching rule is usually applied in one of three occasions. The first occasion is to select one of the available vehicles and assign it to a newly created service request. This is a request initiated assignment (RIA). The second occasion is to assign an available vehicle to a service request in the waiting list (i.e., more than one service request has been created before a vehicle becomes available). This situation is called the vehicle initiated assignment (VIA) [Egbelu and Tanchoco 1984]. The last occasion is when the service request waiting list is empty, an available vehicle is dispatched to specified location and the vehicle status is set to idle. This is called an empty vehicle dispatch (EVD). An available vehicle is defined as a vehicle which is not assigned to any service request and either is moving (to a staging area) or stopped (and idle) at some location on the guidepath network. On the other hand, an unassigned and stopped vehicle is specifically labelled as an idle vehicle.

When a loaded vehicle reaches its destination, it must be able to unload. If there is no space available in the deposit station, i.e., BVI is full, then the vehicle must wait, potentially blocking the guidepath and reducing vehicle productivity. The simplest way to avoid this problem is to reserve a space in the buffer. The buffer reservation may be made by the PS when it initiates a service request, or by the AGVS controller when it assigns the request to a vehicle. While this seems an innocuous decision, it is, in fact, a major decision affecting the system control architecture.

When a service request is created with the requirement of a buffer reservation at its destination and the reservation is made by the AGVS controller, then the following situation may occur. At a particular moment, the buffer at the same destination of several active parts (i.e., parts complete their current operations and plan to continue the next operation at another work cell) is full. Later on, many vehicles become idle because no buffer space is free and no service request is being created. Once a space becomes free, the AGVS controller has to decide which active part having this space and then dispatches a vehicle to it. An assignment can be invoked by a buffer space only when the AGVS controller is responsible for the buffer reservation. In other words, besides a service request and an available vehicle, an empty buffer space (at BVI) is the third factor which can trigger a vehicle assignment.

4. SIMULATION CODE GENERATOR

4.1 Framework

The simulation code generator presented in this paper is written in Quick BASIC version 4.5 and generates a SIMAN simulation program. The model frame defines the static and dynamic characteristics of the system, i.e., the system logic. On the other hand, the experimental frame defines the experimental conditions under which the model is run to generate the output data. Once these frames compiled, linked, and executed in sequence, a summary output report is generated to show the measurements of the system performance. By evaluating these measurements, designer may change the system design and runs the SCG iteratively until a proper system design is obtained. The SCG framework is given in Figure 4 [Haddock 1987].

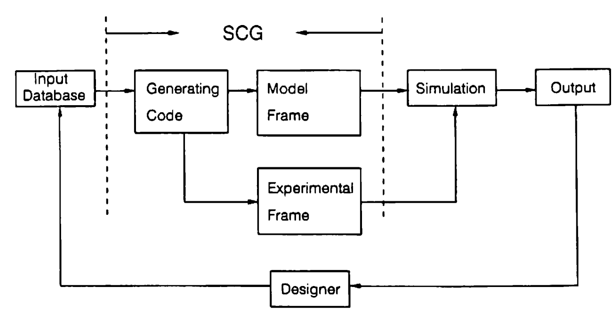

Figure 4. The SCG Framework

4.2 Input Data

The input data contained in 11 files fall in two categories: geometric data and non-geometric data [Bakkalbasi and Durrence 1989]. The geometric data consist of the guidepath network and the location of P/D stations. Additionally, the non-geometric data include vehicle type specifications, workstation characteristics, part process routes, etc. In the following, these 11 files are introduced in detail.

The first file contains a record for each control point information giving the control point identification (ID), location coordinates, and control point type. The information about the guidepath segments is contained in File 2 which includes segment label (ID), length, type, and IDs of its end control points. File 3 consists of a shortest path table (matrix) between each pair of control points. Each element in this matrix specifies the next intermediate control point to travel from one given control point (row) to another (column).

File 4 associated with the static information of vehicle types contains normal speed, acceleration/deceleration, loaded speed adjustment factor, curve speed adjustment factor, load/unload time, and the load capacity. The relation between a vehicle label and its corresponding vehicle type is specified in File 5.

There are five files used to describe the work cell, part type, and process related information. In File 6, P/D station information is presented, specifically its type, buffer size, associated work cell label, and the corresponding control point ID. File 7 contains the static work cell information involving work cell label, capacity, mean time to failure, mean time to repair, and all the P/D station IDs in its domain. File 8 includes information on each part type: part type ID, new part interarrival time distribution, and a pointer to its corresponding process route. The file 9 contains a list of work cells, and the setup and processing times. Based on the pointer specified in File 8 and the work cell list contained in File 9, we can determine the process route of each part type and the corresponding operation time at each work cell in the route. The aggregated flow information between pairs of P/D stations is given in File 10. Finally, File 11 associates all of the file types described above with their corresponding DOS file name and specifies the simulation time unit and total simulation time.

4.3 SCG Structure

As shown in Figure 5, the SCG is formed by three main components: input interpreter (reader), program base, and the model constructor. The input interpreter reads in the external database, arranges data files, and saves data in internal matrices. The program base stores the simulation model frame related program modules (e.g., subroutines, functions, or a set of program statements). Based on the internal matrices, the model constructor retrieves the required program modules from the program base and assembles them into a simulation program.

4.3.1 Program Base

The program base consists of a set of program modules which can be classified into two types: system logic type and the control logic type. The first type modules define the logic of the system (e.g., vehicle move process), which is fixed for all cases within the problem domain.

A process-orientated approach is applied to describe the simulation model logic, in which parts and vehicles are treated as entities. Machines, buffers, and control points are treated as resources. From the part flow viewpoint, process routines are defined to describe the processes undergone by each entity. A rough-cut simulation flow chart is represented in Figure 6. Each block in Figure 6 is composed of a set of SIMAN language statements and is also indicated as a block module.

Interarrival time distributions are used to model the introduction of new parts into the system. Each part type is assumed to have a fixed process route. Depending on their own routes, new parts can be introduced into the system at different work cells.

After a part is inducted successfully, or after it reaches an input buffer (BVI + BMI), it is put in a queue to wait for an available machine. The available machine is assigned to a part according to a selection rule (e.g., FCFS). The part is then delayed for a period of time, representing processing. If there exists an additional processing operation required by the part, it must occupy a position in the output buffer (BVO + BMO) before it releases the machine. After the part enters an output buffer, it has to reserve an input buffer space at its next operation's work cell and then requests a vehicle delivery. As all the required operations have been completed, a part is sent to a special module for collecting statistics and leaves the system.

A vehicle travels between any two locations undergoing a series of movements through the guidepath segments (i.e., linkages of adjacent control points) based on a shortest path analysis. As in [Davis 1986], in order to continue the movement on the next segment, a vehicle must seize the end control point of the segment. Traffic congestion occurs when two or more vehicles try to seize the same control point on the guidepath network at the same time. The congestion also may occur when a vehicle is blocked by another vehicle in front of it. The FCFS rule is used to release a congestion. Therefore, a blocked vehicle has to wait until the congested segment is clear before it can continue its journey.

The second type program modules denote the system control policies for determining decisions of vehicle assignment, path selection, and PS related control (e.g., machine selection rule). Each module is in a function subroutine form. With a set of input data, each module (function) returns the control decision. The data types of input parameters and output variable are deterministic. For future expansion, a new control rule can be added easily just by changing the content of the corresponding module.

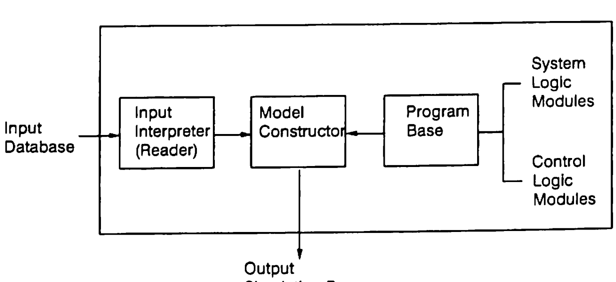

Figure 5. The SCG Structure

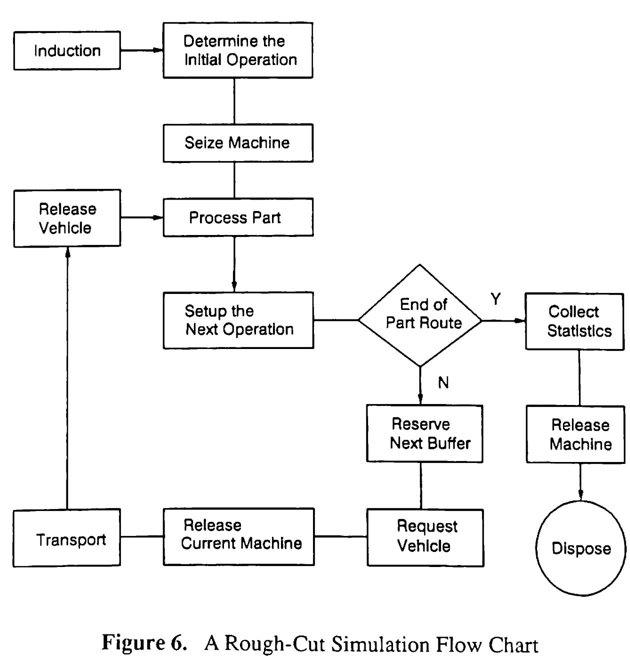

Figure 6. A Rough-Cut Simulation Flow Chart

4.3.2 Model Constructor

Based on the internal matrices, the model constructor recognizes the problem specification. In other words, a set of if-then statements which form a decision tree are contained in the model constructor. Any tree branch provides a certain information about the required program modules corresponding to the problem. The model constructor then retrieves system logic modules together with required modules corresponding to the control policy from the program base to form the simulation model frame. The model constructor is also responsible for generating the simulation experimental frame. In SIMAN, each experimental frame statement is used at most one time. In the similar manner, as a librarian, the model constructor can directly and sequentially generate the required statements by referring the internal matrices to obtain the final experimental frame.

4.4 Simulation Output

The output measurements provides information to evaluate the system performance and to select a proper design. These measurements include the throughput and the average flow time of each part type, work cell utilization, buffer utilization consisting of the average and the maximum queue length, individual and the aggregate vehicle utilization, the loaded and unloaded travel time covering the average, minimum, and the maximum values, as well as the utilization of each control point which can be used to identify the heavy traffic region.

5. EXAMPLE

The system configuration for this example is shown in Figure 7. This system consists of 8 work cells, 1 receiving dock, 1 vehicle staging area, and 36 control points. Three part types are produced in this system and 8 vehicles are used. Table 1 displays the work cell information including work cell label, capacity, and buffer information. In Table 2, the part induction distributions are presented. The part routes and processing times are given in Table 3. Table 4 shows the vehicle specification. Finally, the simulation environment information such as the simulation length, time unit, date, project title, and the analyst name is illustrated in Table 5. Note that these tables present the required data which are capable of describing a system but in a different format than that processed by SCG.

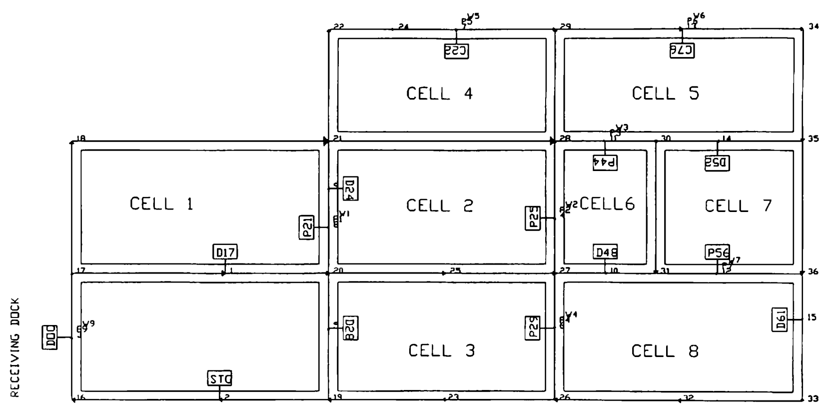

Figure 7. The System Configuration of a Numerical Example

Table 1. Work Cell Information

Work Cell label	capacity	Pickup Station label	capacity	Deposit Station label	capacity	Combined Station label	capacity
CELL1	2	P21	5	D17	5		
CELL2	3	P25	4	D24	4		
CELL3	5	P29	6	D28	6		
CELL4	5					C22	4
CELL5	10					C76	7
CELL6	4	P44	4	D48	4		
CELL7	6	P56	10	D52	10		
CELL8	10			D61	20		
RECDOCK	5					DOC	100
STAGING	10					STG	10

Table 2. Part Induction Distributions

Part type	Interarrival distribution	Parameters (min)
PT01	Exponential	5
PT02	Normal	(5,2)
PT03	Deterministic	10

Table 3. Part Routes and Processing Times

Part type	op1	proc time	op2	proc time	Part route op3	proc time	op4	proc time	op5	proc time	op6	proc time
PT01	RECDOCK	1	CELL3	3	CELL5	4	CELL7	5	CELL6	3	CELL8	1
PT02	RECDOCK	1	CELL4	4	CELL5	5	CELL8	1				
PT03	RECDOCK	1	CELL1	5	CELL2	5	CELL8	1				

Table 4. Vehicle Specification

No. of veh in system	Speed (ft/min)	Load/unload time (min)
8	180	0.25

Table 5. Profile

Simulation time length	Time unit	Date	Project title	Analyst name
480	min	1-1-1990	Demo	Tester

The SIMAN summary report for this example is shown in Table 6. It includes a title and two subreports which display statistics for part flow time, vehicle utilization, control point utilization, and work cell utilization. From the tally variables subreport, the average unloaded travel time is 1.79 minutes and the average loaded travel time is 2.11 minutes. There were 103 type 1 parts processed during 480 minutes. The average processing time for part type 1 was 59.68 minutes. Similarly, statistics are given for part types 2 and 3. In the discrete change variables subreport, the first row shows an average of 7.88 vehicles busy, which corresponds to a utilization of 96%. The utilizations of work cells and their buffers are drawn from the data beneath the first row. The "AGV REQST QUEUE" displays an average length of 8.93 parts in the line waiting to request an available vehicle. On average, 0.37 parts were waiting for a buffer space. Next, the individual vehicle utilization is given. The control points occupancy utilizations are displayed at the end.

6. LIMITATIONS

In this section, the limitations in use the current version simulation code generator are discussed. In other words, the SCG creates feasible simulation programs that have to meet the following restrictions.

(1) Each work cell contains one or a group of identical machines. The work cell also has one P and/or one D, or one combined station which has the function as the pickup and the deposit stations.
(2) The process route of each part type is deterministic. Multi-route cases are not considered.
(3) Buffer reservation is required before requesting a vehicle.
(4) First-come-first-serve (FCFS) rule is used to select a part from the buffer for the next operation (i.e., machine selection rule).
(5) Vehicles with a constant speed travel along the guidepath. The travel time on a guidepath segment equals the segment length divided by the vehicle speed.
(6) Vehicles follow the shortest path traveling between work cells.
(7) At most only one vehicle can occupy a segment at one time.
(8) The traffic congestion is released under the FCFS rule.
(9) The idle vehicle is sent back to the staging location when no more requests exist. Otherwise, this idle vehicle waits to load an outbound part from the prior request destination if there exists one or is assigned to the oldest service request.
(10) There is only one type of the vehicle in the system.
(11) No breakdown is considered.

Table 6. The SIMAN Report

Tally Variables

Number Identifier	Average	Standard Deviation	Minimum Value	Maximum Value	Number of Obs.
Unloaded Trav Time	1.79733	1.44841	.00000	7.19900	986
Loaded Trav Time	2.11549	1.45014	.11666	6.30551	962
PT01 Flow Time	59.68442	10.30560	44.71615	88.52524	103
PT02 Flow Time	33.98848	9.41856	21.77005	57.19513	85
PT03 Flow Time	34.11152	6.09358	20.96311	47.24710	44

Discrete Change Variables

Number Identifier	Average	Standard Deviation	Minimum Value	Maximum Value	Time Period
AGGR AGV UT	7.88187	.78866	.00000	8.00000	480.00
CELL1 UT	.48631	.50206	.00000	2.00000	480.00
CELL2 UT	.47471	.50641	.00000	2.00000	480.00
RECDOCK UT	.55534	.72747	.00000	4.00000	480.00
DLV17 UT	.00010	.00992	.00000	1.00000	480.00
PK21 UT	.50429	.50920	.00000	2.00000	480.00
CMB76 UT	1.66751	1.22706	.00000	5.00000	480.00
DOCK UT	5.14710	3.18029	.00000	18.00000	480.00
AGV REQST Q	8.93301	5.86338	.00000	24.00000	480.00
WAIT4 BUFF Q	.37730	.96521	.00000	7.00000	480.00
VEH101 STATUS	.95457	.20824	.00000	1.00000	480.00
VEH102 STATUS	.95508	.20713	.00000	1.00000	480.00
CTRL PT 1 UT	.17371	.37886	.00000	1.00000	480.00
CTRL PT 2 UT	.19374	.39523	.00000	1.00000	480.00
CTRL PT 23 UT	.51431	.49980	.00000	1.00000	480.00
CTRL PT 36 UT	.25031	.43319	.00000	1.00000	480.00

7. SUMMARY AND FUTURE WORK

In this paper, a simulation code generator for AGVS design has been presented. A process-oriented approach is used to describe the simulation model logic. However, this approach does not explicitly address the AGVS operational concept which was discussed in Section 3. Therefore, we currently develop a new system logic diagram in a network form. As shown in Figure 8, each node corresponds to an event or a process. Nodes 1 to 6 express the AGVS related activities. The production system (PS) activities are modelled in nodes 7 to 12. The detailed description of each node is given in Table 7.

Sixteen different types of functions will be used in model the entire control policy. Four functions are dealing with the vehicle dispatching. Two functions concern the vehicle navigation path determination and one function resolves the traffic congestion caused by seizing the same control point simultaneously.

The others are PS related functions. They are used to determine the next operation work station; the transfer times between buffers or between buffer and machine at the same work cell; machine selection rule; buffer selection rule; blocked machine selection rule; or the buffer reservation respectively.

Furthermore, two types of vehicle travel times moving on a guidepath segment are discussed. As displayed in Figure 9, the type 1 travel time denotes the move from the beginning of a segment to the internal check point. The type 2 travel time presents the move from the internal check point to the end of the segment. If a vehicle is allowed continuously to move to the next segment without stopping, it will pass the internal check point of the current segment in a normal speed. Otherwise, the vehicle has to reduce the speed from the internal check point and has to fully stop at the end of the current segment.

Table 7. Network Diagram Description

Node	Description
1	Assign task
2	Check next control point availability
3	Begin a segment travel
4	Begin blocking
5	End a segment travel
6	End load/unload
7	Induction
8	Determine the next operation
9	End the machine load/unload
10	End machining
11	End the buffer transfer
12	Collect statistics

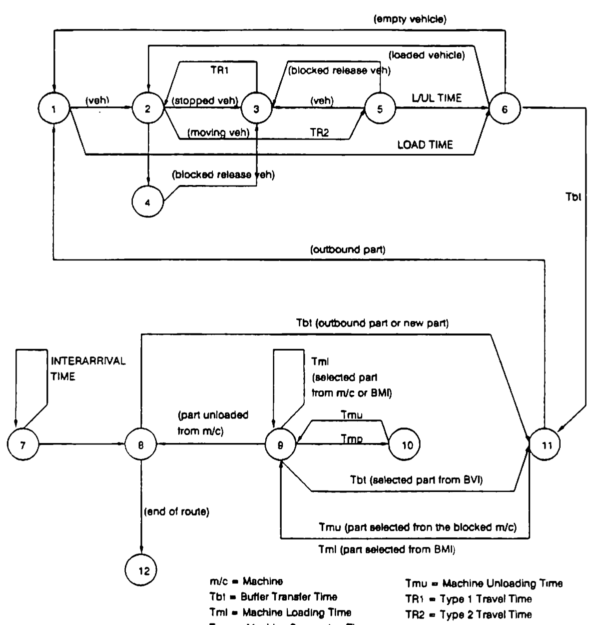

Figure 8. A System Logic Network Diagram

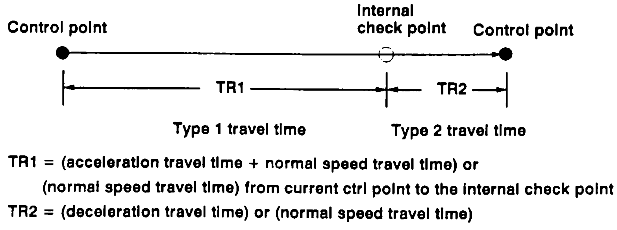

Figure 9. Two Types of Travel Times on a Segment

The future work is to implement this network diagram in a new SCG. This new SCG provides the potentials of multiple P/D stations in a work cell, multiple process route of a part type, and several different control policies. Additional control rules written by designers are also capable of being linked with the SCG.

ACKNOWLEDGMENTS

This paper is based on work partially supported by NSF Grant CDR-8300965, the Georgia Institute of Technology, and the Member Companies of the Material Handling Research Center.

REFERENCES

Bakkalbasi, O. and G. Durrence (1989), "An Engineering Workstation for AGVS Design: User's Manual," Material Handling Research Center, Georgia Tech, Atlanta, Georgia.

Balmer, D.W. (1987), "Modelling Styles and Their Support in the CASM Environment," In *Proceedings of the 1987 Winter Simulation Conference*, A. Thesen, H. Grant, and W.D. Kelton, Eds. IEEE, Piscataway, NJ, 478-486.

Clementson, A.T. (1986), "Simulation with Activities Using C.A.P.S./E.C.S.L. (The British Approach to Discrete-Event Simulation)," In *Proceedings of the 1986 Winter Simulation Conference*, J.R. Wilson, J.O. Henriksen, and S.D. Roberts, Eds. IEEE, Piscataway, NJ, 113-122.

Davis, D.A. (1986), "Modeling AGV Systems," In *Proceedings of the 1986 Winter Simulation Conference*, J.R. Wilson, J.O. Henriksen, and S.D. Roberts, Eds. IEEE, Piscataway, NJ, 568-574.

Egbelu, P.J. and J.M.A. Tanchoco (1984), "Characterization of Automatic Guided Vehicle Dispatching Rules," *International Journal of Production Research 22*, 3, 359-374.

Gaskins, R.J. and J.M.A. Tanchoco (1989), "AGVSim2 – A Development Tool for AGVS Controller Design," *International Journal of Production Research 27*, 6, 915-926.

Haddock, J. (1987), "An Expert System Framework Based on a Simulation Generator," *Simulation 48*, 2, 45-53.

Mathewson, S.C. (1985), "Simulation Program Generators: Code and Animation on a P.C.," *Journal of the Operational Research Society 36*, 7, 583-589.

Miller, R.K. (1987), *Automated Guided Vehicles and Automated Manufacturing*, Society of Manufacturing Engineers, Dearborn, MI.

Pegden, C.D. (1987), *Introduction to SIMAN*, Systems Modeling Corporation, State College, PA.