IV. SIMULATION-BASED PERFORMANCE MODELS

Annals of Operations Research 3(1985)171 – 188

A GPSS/H MODEL FOR A HYPOTHETICAL FLEXIBLE MANUFACTURING SYSTEM

T.J. SCHRIBER

Graduate School of Business Administration, University of Michigan,
Ann Arbor, Michigan 48109, USA

Abstract

A GPSS/H model is presented for a hypothetical flexible manufacturing system. The FMS consists of six machines composed of three machine types, manufactures three types of parts, and uses automatic guided vehicles (AGVs) to transport in-process parts between appropriate machines and wait spaces in the system. Three logical modules have been designed for the model, with copies of these modules then being appropriately distributed and interfaced throughout the model and tailored to achieve overall representation of the specific FMS. The same technique can be used by others to build analogous or extended GPSS/H models for other specific FMSs in which AGVs are used as transporters. Simulations can then be performed with such models to research FMS design and control alternatives.

Keywords and phrases

Automatic guided vehicles, discrete-event simulation, flexible manufacturing systems, GPSS, GPSS/H, production rates.

1. Introduction

Computer-based simulation modeling provides a viable means of experimenting with alternative designs of flexible manufacturing systems (FMSs). Decisions which must be made in the design of such systems include the types and numbers of machines to include in the system, the number of wait spaces to provide for in-process parts waiting to acquire the type of machine they need next, the relative placement of these machines and wait spaces with respect to each other, and the number of automatic guided vehicles (AGVs) to have in the system. Simulation also provides a means of experimenting with alternative policies and procedures for controlling the operation of flexible manufacturing systems. Included among such policies are criteria for admitting new work to the system, the maximum level of in-process inventory to permit in the system, and rules for deciding which of two or more waiting parts to send next to an available machine for which they are competing. Design, planning, scheduling, and control problems of flexible manufacturing systems are discussed in [9]. Likewise, the role of simulation modeling in addressing such problems is discussed in [7].

A decision to engage in computer-based simulation modeling leads to the need to choose a language for use in the modeling process. A recent survey [1] indicates that a large number of simulation practitioners still choose the programming language FORTRAN for this purpose, whereas others, fewer in number, choose modeling-oriented languages such as GPSS, SLAM, SIMAN, or SIMSCRIPT. Among the compelling reasons for choosing a modeling language such as GPSS are these (quoted from Henriksen and Crain, ref. [5], p. 2-1):

> 'A. By providing for automatic collection and output of statistics, GPSS relieves the user of responsibility for specifying tedious programming details such as output formats.
> B. GPSS provides extensive run-time error detection. If a simulation were to be built 'from scratch' in another language, the incorporation of such checks would be a major burden.
> C. Some of the algorithms employed by the GPSS simulator are very sophisticated and provide capabilities that could not easily be achieved if a simulation were to be built 'from scratch'. For an example, see reference (Henriksen, 1977 [3]).'

This paper reports the results of a project undertaken to demonstrate the relative ease and utility of using GPSS to model flexible manufacturing systems. The demonstration takes the form of building and using a GPSS model for a hypothetical FMS. A certain degree of modularity is achieved in building the model. This not only facilitates the building of the model discussed here, but also provides a pattern others may find useful in building GPSS models for other specific FMSs.

Section 2 of the paper provides an overview of GPSS, and includes a brief example of a simple GPSS model and shows some of the output which results from use of the model. Section 3 describes the hypothetical FMS chosen to demonstrate the use of GPSS/H in modeling such systems. Section 4 discusses some of the decisions made in choosing the various GPSS modeling entities used to represent various aspects of the FMS. Section 5 provides a glimpse of the resulting GPSS/H model, and sect. 6 comments on some of the output produced by the model. Brief comment on potential applications of the model are provided in sect. 7, and sect. 8 speculates on taking the ideas incorporated into the model and applying them to the modeling of other specific FMSs. The paper concludes with a summary, acknowledgements and references.

2. An overview of GPSS

GPSS (general purpose simulation system) is a highly popular simulation language which lends itself especially well to the modeling of systems in which discrete units of traffic compete for scarce resources. GPSS was originally released by IBM in 1961. It subsequently evolved through a series of IBM releases (GPSS II; GPSS III; GPSS/360; and, in 1970, GPSS V), each offering enhancements over its predecessor. Paralleling the IBM releases, a variety of GPSS implementations was made available both for IBM and non-IBM hardware by organizations external to IBM. The state-of-the-art GPSS implementation for IBM mainframe hardware is now GPSS/H, which became available in 1977 and is an upwardly compatible superset of IBM's GPSS V [5].

Among the more significant advantages offered by GPSS/H over GPSS V are an improvement in execution speed by a factor of about 5 on average; the ability to interactively monitor an ongoing simulation, which greatly reduces the time required to build and debug models and achieve a detailed understanding of their behaviour; the ability to read from and write to external files, which facilitates the incorporation of data into models and the passing of model outputs to post-processing software, such as graphical routines; the use of long symbolic names in extended contexts, which enhances model readability and clarity; and vastly improved ease of accessing FORTRAN subroutines and functions during an ongoing GPSS simulation.

GPSS/H is also available for VAX computers, for the IBM AT/370, and for selected microcomputers based on the Motorola 68000 chip. Another VAX implementation of GPSS is GPSS/VAX [6], and GPSS/PC exists for the IBM PC [2]. The survey of simulation practitioners cited earlier indicates that GPSS is in very widespread use [1].

In the GPSS world view, or stylized way of looking at a problem, units of traffic (which are called transactions) are visualized as moving from point to point in a system, obtaining and using constrained resources as they move. For example, a transaction might be a part moving through a manufacturing system, and each point it moves through in the system might be a machine of a specified type. As a part (transaction) moves from point to point (machine to machine), it is transformed over time into a finished part.

The GPSS world view provides a natural, convenient, and powerful way to model a wide variety of systems, and is a major reason why GPSS has become a classic language. GPSS is also versatile enough to support alternative approaches to the modeling of systems (see [4]).

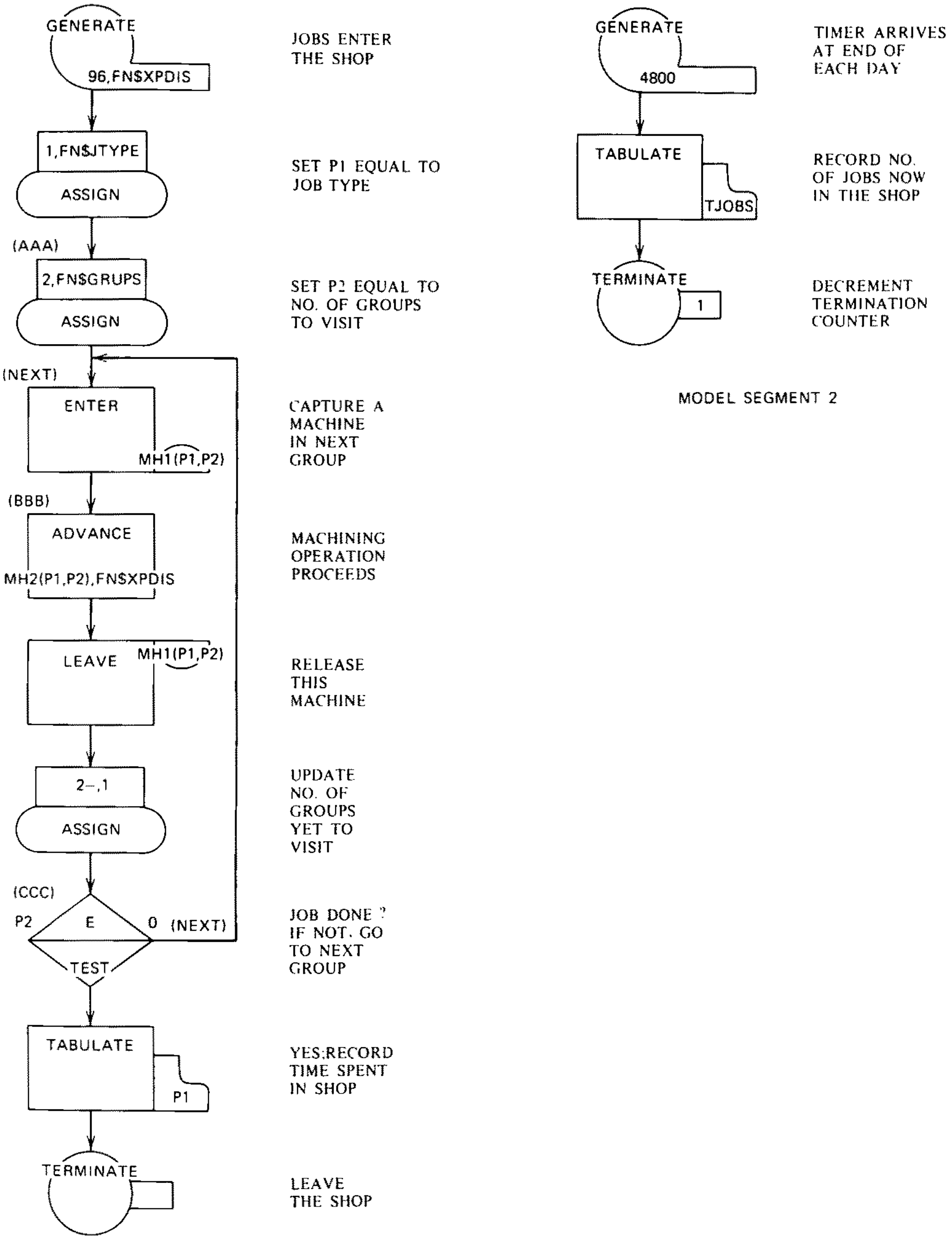
In a manufacturing system, machines are constrained resources for which parts must compete. The GPSS language provides entities which can be used to represent such constrained resources. (The facility entity is used to model unit resources in GPSS; the storage entity is used to model pools of resources of identical type.) A variety of queue disciplines (e.g. first-come, first-served; shortest processing time;

fewest remaining steps before completion; longest time in system; and so on) can be modeled in GPSS to resolve conflicts when contention for constrained resources occurs. Resources modeled with facility and storage entities can be subjected to periods of unavailability (to reflect such things as periodic machine maintenance, or machine breakdowns), and resources modeled with the facility entity can be subjected to preemption (that is, the current user of such a resource can have its ongoing use of the resource interrupted on behalf of another user).

Use of facility and storage entities results in automatic collection and automatic postsimulation display of statistics associated with these entities (e.g. fraction of time in use; capture count; average holding time per capture; and, for the storage entity, maximum number of resource units in use simultaneously). GPSS also offers additional types of entities which can be used to enhance the statistical information produced in a simulation. For example, the queue entity can be used to collect information about units of traffic waiting to capture constrained resources (e.g. average waiting time; average number waiting; maximum number waiting; number of traffic units which did not have to wait at all). And the table entity can be used to tabulate observed values of random variables of interest, such as time-in-system, producing such things as sample average; sample standard deviation; frequency class counts; relative frequencies; and cumulative frequencies. By providing for such automatic collection and output of statistics, GPSS enhances the statistical insights which a simulation provides about the characteristics and behavior of the system being modeled.

GPSS offers a collection of types of blocks which the modeler arranges in appropriate ways in block diagrams to express the rules governing the operation of the system being modeled. Each block can be thought of as a point at which an underlying subroutine (automatically provided by GPSS) is called, with a subroutine call occurring whenever a unit of traffic moves into a block. Blocks are provided for such functions as introducing units of traffic into a model; having a unit of traffic issue a request for a unit resource being modeled with a facility; or for one or more units of resource being modeled with a storage; having a unit of traffic relinquish control of previously captured resources; observing and tabulating values of user-specified random variables; checking in to or out of waiting lines; being removed from a model; and so on. In total, some 45 different types of blocks are available in GPSS, providing a rich set of modeling capabilities. An executing GPSS model can also invoke FORTRAN subroutines (and, in GPSS/H, FORTRAN functions as well) to accomplish certain effects which might lend themselves more naturally to a FORTRAN environment than to the immediate environment of GPSS.

A GPSS model can take the form either of a block diagram, or of the statements corresponding to a block diagram. (A block diagram must be re-expressed in statement form before a corresponding simulation can be performed.) A typical GPSS block diagram (Schriber, ref. [8], pp. 229 et. seq.) is shown in fig. 2.1. This block diagram models a production system consisting of six machine groups which are used



MODEL SEGMENT 2

MODEL SEGMENT 1

Fig. 2.1. GPSS block diagram for a simple production system.

```
       SIMULATE


*
*      FUNCTION DEFINITION(S)
*
 GRUPS FUNCTION   P1,D3         NO. OF MACHINE GROUPS EACH JOB TYPE VISITS
1,4/2,3/3,5
 JTYPE FUNCTION   RN1,D3        DISTRIBUTION OF JOB-TYPES
.24,1/.68,2/1,3
 XPDIS FUNCTION   RN1,C24       EXPONENTIAL DISTRIBUTION FUNCTION
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38
.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2
.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
*
*      MATRIX SAVEVALUE DECLARATION(S)/INITIALIZATION(S)
*
 1     MATRIX     H,3,5         MATRIX OF VISITATION SEQUENCES
       INITIAL    MH1(1,1),6/MH1(1,2),2/MH1(1,3),3/MH1(1,4),1
       INITIAL    MH1(2,1),2/MH1(2,2),4/MH1(2,3),5
       INITIAL    MH1(3,1),6/MH1(3,2),3/MH1(3,3),4/MH1(3,4),5
       INITIAL    MH1(3,5),1
*
 2     MATRIX     H,3,5         MATRIX OF MEAN MACHINING TIMES
       INITIAL    MH2(1,1),600/MH2(1,2),200/MH2(1,3),350/MH2(1,4),1250
       INITIAL    MH2(2,1),650/MH2(2,2),900/MH2(2,3),1050
       INITIAL    MH2(3,1),250/MH2(3,2),300/MH2(3,3),500/MH2(3,4),2500
       INITIAL    MH2(3,5),2350
*
*      STORAGE CAPACITY DEFINITION(S)
*
       STORAGE    S1,14/S2,5/S3,4   PROVIDE 14, 5, AND 3 MACHINES IN
*                                   GROUPS 1, 2, AND 3, RESPECTIVELY
       STORAGE    S4,8/S5,16/S6,4   PROVIDE 8, 16, AND 4 MACHINES IN
*                                   GROUPS 4, 5, AND 6, RESPECTIVELY
*
*      TABLE DEFINITION(S)
*
 1     TABLE      M1,2400,2400,10   SHOP RESIDENCE TIME (TYPE 1 JOBS)
 2     TABLE      M1,2400,2400,10   SHOP RESIDENCE TIME (TYPE 2 JOBS)
 3     TABLE      M1,2400,2400,10   SHOP RESIDENCE TIME (TYPE 3 JOBS)
 TJOBS TABLE      V$COUNT,10,10,5   TOTAL JOBS IN THE SHOP
*
*      VARIABLE DEFINITION(S)
*
 COUNT VARIABLE   W$AAA+W$BBB+W$CCC TOTAL JOBS IN THE SHOP
*
*      MODEL SEGMENT 1
*
       GENERATE   96,FN$XPDIS         JOBS ENTER THE SHOP
       ASSIGN     1,FN$JTYPE          SET P1 = JOB TYPE
 AAA   ASSIGN     2,FN$GRUPS          SET P2 = NO. OF GROUPS TO VISIT
 NEXT  ENTER      MH1(P1,P2)          CAPTURE A MACHINE IN NEXT GROUP
 BBB   ADVANCE    MH2(P1,P2),FN$XPDIS MACHINING OPERATION PROCEEDS
       LEAVE      MH1(P1,P2)          RELEASE THIS MACHINE
       ASSIGN     2-,1                UPDATE NO. OF GROUPS YET TO VISIT
 CCC   TEST E     P2,0,NEXT           JOB DONE? IF NOT, GO TO NEXT GROUP
       TABULATE   P1                  YES;  RECORD TIME SPENT IN SHOP
       TERMINATE                      LEAVE THE SHOP
*
*      MODEL SEGMENT 2
*
       GENERATE   4800                TIMER ARRIVES AT END OF EACH DAY
       TABULATE   TJOBS               RECORD NO. OF JOBS NOW IN THE SHOP
       TERMINATE  1                   DECREMENT TERMINATION COUNTER
*
*      CONTROL CARDS
*
       START      5                   START RUN FOR WEEK 1
       RESET                          ZERO-OUT ACCUMULATED STATISTICS
       START      5                   START RUN FOR WEEK 2
       RESET                          ZERO-OUT ACCUMULATED STATISTICS
       START      5                   START RUN FOR WEEK 3
       RESET                          ZERO-OUT ACCUMULATED STATISTICS
       START      5                   START RUN FOR WEEK 4
       RESET                          ZERO-OUT ACCUMULATED STATISTICS
       START      5                   START RUN FOR WEEK 5
       END                            RETURN CONTROL TO THE SYSTEM
```

Fig. 2.2. Model file corresponding to the fig. 2.1 block diagram.

to manufacture three types of product. Each product type is characterized by its own machine visitation sequence and its own machining time requirements. Queue discipline used at the machine is first-come, first-served. Transactions are used to simulate jobs moving through the system, the storage entity is used to model the six machine groups, and the table entity is used to collect information about job residence time in the shop (as a function of job type), and the total number of jobs in the shop. Figure 2.2 shows the same model in statement form where, in addition to the statements corresponding to blocks, several GPSS functions are defined, several matrices are declared and populated with INITIAL statements (the matrices contain information about the machine visitation sequences and the machining time requirements), and various simulation control statements are included.

```
STORAGE     CAPACITY    AVERAGE       AVERAGE        ENTRIES     AVERAGE      CURRENT     MAXIMUM
                        CONTENTS    UTILIZATION                 TIME/TRAN    CONTENTS    CONTENTS
      1           14     10.074        .719             136     1777.830         14          14
      2            5      2.846        .569             141      484.560          3           5
      3            4      1.724        .431             110      376.299          4           4
      4            8      4.432        .554             150      709.166          2           8
      5           16     10.657        .666             166     1540.843         16          16
      6            4      1.703        .425             106      385.716          3           4


TABLE  TJOBS
ENTRIES IN TABLE        MEAN ARGUMENT          STANDARD DEVIATION        SUM OF ARGUMENTS
              5                38.799                      11.625                194.000         NON-WEIGHTED

       UPPER       OBSERVED      PER CENT      CUMULATIVE      CUMULATIVE      MULTIPLE       DEVIATION
       LIMIT      FREQUENCY      OF TOTAL      PERCENTAGE       REMAINDER       OF MEAN       FROM MEAN
          10              0           .00              .0           100.0          .257          -2.477
          20              0           .00              .0           100.0          .515          -1.617
          30              2         39.99            39.9            60.0          .773           -.756
          40              1         19.99            59.9            40.0         1.030            .103
    OVERFLOW              2         39.99           100.0              .0
  AVERAGE VALUE OF OVERFLOW          50.50
```

Fig. 2.3. Selected output from the production system model.

Selected output (automatically produced, and of fixed form and content) from the production system model appears in fig. 2.3. Storage statistics appear in the upper part of this figure. (For example, storage 1 models a pool of 14 machines of a given type. On average, 10.074 of these machines were in use, with a corresponding utilization of 0.719. There were 136 captures of this type of machine. Average holding time per capture was 1777.83 time units. When the statistics were produced, all 14 of the machines were in a state of capture. The maximum number of these machines captured simultaneously matches the total number of these machines, 14.) The table information corresponding to observations taken on the number of jobs in the system appears in the lower part of fig. 2.3. (For example, based on five observations there were 38.799 jobs in the system on average, with a standard deviation of 11.625.)

The simple model in fig. 2.1 is offered here to provide a quick feeling for the character of GPSS, with no attempt having been made to explain the details of the model. Furthermore, this is not a model of a flexible manufacturing system. Consider, for example, that the use of fixtures and pallets in the system, and the shortest-path transporting of parts between machines and through queuing areas with automatic guided vehicles, are not explicitly represented in this simple model.

3. Characteristics of the hypothetical FMS

The flexible manufacturing system chosen here for modeling in GPSS/H is shown in fig. 3.1, where the physical layout of the system is depicted. The system is composed of a rectangular grid consisting of six rows and three columns. Each of the 18 points in the grid is shown as a box in fig. 3.1. These boxes are connected by straight line segments representing paths along which AGVs can move. Each point,

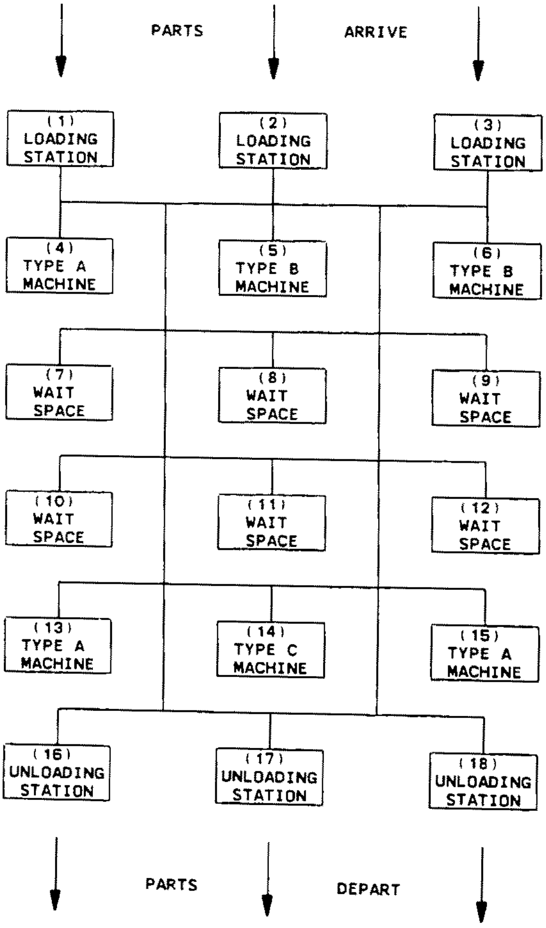


Fig. 3.1. Physical layout of the hypothetical FMS.

or box, is a location at which a unit resource is located. The various alternative unit resources are machines, wait spaces, loading stations (points at which incoming parts are put into fixtures and onto pallets), and unloading stations (points at which finished parts are removed from their pallets and fixtures and from the system).

There are 3 type A machines in the system, 2 type B machines, and 1 type C machine. There are 6 wait spaces. The system is used to manufacture parts of types 1, 2 and 3. Each part type has its own dedicated loading station and its own dedicated unloading station.

The boxes in fig. 3.1 have been numbered 1 through 18. Boxes 1, 2, and 3 are the loading stations for parts of type 1, 2, and 3, respectively. Boxes 4, 5, and 6 are machines. (Box 4 is a type A machine, boxes 5 and 6 are type B machines.) Boxes 7 through 12 are wait spaces. Boxes 13, 14, and 15 are machines. (Boxes 13 and 15 are type A machines, box 14 is the type C machine.) Finally, boxes 16, 17, and 18 are the respective unloading stations for part types 1, 2, and 3.

Other system resources not explicitly shown in fig. 3.1 are fixtures, pallets, and carts (AGVs). Each type of part is assumed to require its own type of fixture, but parts can make use of any pallet and any cart when these resources are needed and available.

Parts of type 1 visit machines of type A, B, and C, in that order, with machining time requirements of 180, 120, and 75 min, respectively. Parts of type 2 visit machines of type B and A, in that order, with machining time requirements of 90 and 150 min, respectively. Parts of type 3 simply visit machine A, and have a machining time requirement there of 300 min.

Fifty percent of the parts manufactured are of type 1, thirty percent of type 2, and the remaining twenty percent are of type 3.

A fixed number of parts of mixed type is admitted to the system initially. In this model, this number can not exceed one less than the sum of the number of machines and the number of wait spaces. Otherwise, deadlock might come about. (Deadlock is a situation in which no part in the system can move.) This could occur, for example, if all wait spaces were filled, and if all machines were occupied by parts which needed to visit at least one more machine before being finished.

When a finished part leaves the system, another part is admitted to the system. The admitted part may differ in type from the leaving part. Parts are admitted in the part-type sequence 1, 3, 1, 2, 1, 3, 1, 2, 1, 2. For example, when the first finished part leaves the system, the next part admitted is of type 1; when the second finished part leaves, the next part admitted is of type 3; when the third finished part leaves, the next part admitted is of type 1, and so on. This admission sequence is cycled through repeatedly over time.

Parts move from point to point in the system according to the following procedure. A part which has just entered the system and has been put into a fixture on a pallet waits in its loading station until both a cart and a machine of the type it first needs become available to it. The part then claims the closest idle cart and the closest idle machine of the correct type, the cart travels to the loading station, picks up the part, and transports it to the machine. The cart then becomes idle, and remains at that machine until it is again needed (either by another part, or by the same part). When finished at that machine, the part then waits at the machine until a cart and either a wait space or a machine of the type it next needs become available to it. The part then claims the closest idle cart, and either the closest idle machine of the correct type, or the closest wait space, whichever applies. The cart travels to the machine, picks up the part (at which time the machine then becomes available again to other parts), and transports it to its next machine, or to the wait space, whichever applies. The cart then becomes idle, and remains where it is until it is again needed (either by another part, or by the same part). If the part has gone to a wait space, the part remains there until both a cart and a machine of the type it needs become available to it, and then proceeds from the wait space to its next machine, and so on.

Eventually, a part is finished at its last machine, and then its 'next machine' is not a machine at all, but is the unloading station dedicated to that type of part. The part waits at its last machine until a cart and the unloading station become available to it. The part then claims the closest idle cart, proceeds to the unloading station, and exits the system, causing another part to be admitted to the system and leaving the cart in idle mode at the unloading station.

Shortest processing time is used in the system to dispatch parts to machines. This means that when a machine and cart become available and two or more part types are waiting for the machine, that waiting part with the shortest processing time on that machine is next to capture the machine. Ties for shortest processing time are resolved first-come, first-served.

Travel times in the system are a linear function of the distance being traveled. These times are set at 1 min per segment traversed. Referring to fig. 3.1, adjacent boxes in a given row are each one segment apart. For example, boxes 4 and 5 are one segment apart, boxes 4 and 6 are two segments apart. Consecutive rows in fig. 3.1 are also each one segment apart. But note that boxes in consecutive rows are a minimum of 2 segments apart. For example, boxes 4 and 7 are two segments apart. (These two segments consist of 1/2 segment to move from box 4 to the path connecting the second and third rows, 1 segment to move from the second to the third row, and 1/2 segment to move to box 7 from the path connecting the second and third rows.) Boxes 4 and 8 are also 2 segments apart, boxes 4 and 9 are 3 segments apart, and so on.

The fixturing of a part takes 5 min. It takes 3 min to transfer a loaded pallet to or from a cart (independent of whether the transfer is from the loading station, to or from a machine, to or from a wait space, or to an unloading station). Fixturing and transfer time are independent of part type.

There are 3 carts in the system, 10 pallets, and 5, 3, and 2 fixtures for parts of type 1, 2, and 3, respectively.

Certain simplifying assumptions (which could be relaxed) have been made about the fig. 3.1 FMS. For example, it is assumed that equipment breakdowns do not occur. (As indicated in sect. 2, GPSS provides specific capabilities for modeling such breakdowns.) Furthermore, the possibility of traffic congestion is ignored. In other words, travel time between two points depends only on the shortest distance between the two points, and does not take into account the possibility that delays will be incurred or an alternative path must be taken because one or more segments along the shortest path are being used by another cart or carts. (Travel times are small compared with machining times, and the probability that two carts are in motion in opposite directions on the same segment simultaneously is small.) And machines start work as soon as possible, rather than being left deliberately idle on occasion while waiting for a relatively high-priority part which is nearing the end of a machining step on some other machine. Finally, the hypothetical FMS does not embody the concept that two or more alternative machine types might be equally suitable for carrying out some machining steps on some part types. These are among the more notable simplifications assumed for the hypothetical FMS.

4. Approach used to model the FMS in GPSS/H

The GPSS/H model for the fig. 3.1 system has been built with facilities, storages, logic switches, queues, Boolean variables, functions, and matrices. Space limitations here do not permit a detailed explanation of the model for readers unfamiliar with GPSS fundamentals. However, a word sketch of the key aspects of the GPSS modeling approach is provided below. This sketch, together with the listing of the model itself, which is heavily commented, will make the model understandable and usable in operational terms for those familiar with GPSS.

Transactions simulate parts moving through the system. Each transaction moves through a model segment corresponding to the type of part it represents. (It would be possible to achieve a higher degree of model compaction by tagging each transaction with a part-type identification and having all transactions move through one and the same model segment, but such compaction would be achieved at the expense of clarity, and this tradeoff is judged to be unfavorable for the purpose at hand.)

Transactions in the model carry parameters which are referred to by a symbolic name. For example, CLOC is the parameter containing a *c*art *loc*ation, MYLOC is the parameter containing the part's current *loc*ation, and PREVMLOC is the parameter containing the *loc*ation of the *prev*ious *m*achine (if any) used by the part. This use of symbolic parameter names (available in GPSS/H and GPSS/PC) greatly enhances the clarity and readability of the model.

Each of the boxes in fig. 3.1 is modeled with a facility corresponding to the unit resource which the box represents. Facilities 1 through 18 are used to model boxes 1 through 18, respectively. When a unit resource is in use, the corresponding facility is in use, and when idle, the corresponding facility is idle. By examining this logical attribute of a facility of interest, a transaction can determine if the corresponding unit resource is available to it or not.

The facilities in the model have been *equ*ivalenced to mnemonically chosen symbolic names, making it easy to identify the standard GPSS facility statistics of interest in the output. (The software remembers names of symbolically referenced entities, and uses this information to label model output.)

Boxes 4 through 18 in fig. 3.1 are points at which idle carts might be located in the system. (Note that carts would never become idle at boxes 1, 2, or 3, for logical reasons.) Logic switches (on-off switches) 4 through 18 are used in the model to indicate cart availability, or lack thereof, at these corresponding points. When logic switch j is set (j = 4, or 5, or 6, or. . .), then there is an idle cart at the corresponding location; when a logic switch is reset, this means there is no idle cart at the corresponding location.

Groupings of common resource units, such as the 3 type A machines, the 2 type B machines, and the 6 wait spaces, are modeled with storages. This means that some resources are modeled both individually (with the facility entity) and collectively (with the storage entity). Furthermore, the pool of carts is modeled with the storage entity, meaning that cart status is modeled both individually (with the logic switch entity) and collectively. This collective modeling of common pools of resources with the storage entity makes it easy for transactions to test for the availability of needed resources. For example, to test for the availability of a cart and a type A machine, a transaction can evaluate the Boolean expression SNF$CARTS*SNF$MTYPEA, where CARTS is the carts storage and MTYPEA is the storage modeling the pool of type A machines. The Boolean expression is true when at least one cart and one type A machine are idle. When the Boolean expression becomes true, then the transaction (part) can proceed to identify the nearest available cart (by examining the settings of logic switches 4 through 18 in an order from 'closest to my location' to 'furthest from my location') and the nearest available type A machine (by examining the status of the type A machine facilities, again in an order from 'closest to my location' to 'furthest from my location').

Matrices store the scanning sequences used by transactions to identify the nearest available cart and the nearest available machine of the correct type. The matrix CLOC (*c*art *loc*ation) consists of 15 columns (one column for each of the 15 locations at which a cart might be needed, with these locations numbered 1 through 15, corresponding to boxes 1 through 15 in fig. 3.1) and 15 rows (one row for each of the 15 locations at which an idle cart might be available, with these locations numbered 4 through 18, corresponding to boxes 4 through 18 in fig. 3.1). For each alternative location at which a cart might be needed, the identically numbered CLOC matrix column stores the candidate starting cart locations, arranged in the column from bottom to top in order of next closest potential starting locations. (Transactions use a 'count down to zero' LOOP block to scan this column in bottom to top order.)

The matix WSLOC is used to store *w*ait *s*pace *loc*ation information, with the arrangement and accessing pattern for this matrix being analogous to that for the CLOC matrix.

The matrix TTIME is used to store the various 'from where, to where' travel times. This matrix consists of 18 rows (one for each potential 'from' location) and 18 columns (one for each potential 'to' location). Strictly speaking, only a triangular matrix is needed for these travel times; for ease of accessing the TTIME matrix, however, a symmetric square matrix has been used.

Analogous to the cart location (CLOC) matrix, matrices store the scanning sequences used by transactions to identify the nearest available machine of the correct type. There is one such matrix for each combination of a part type and a next machine type needed. Rows correspond to the alternative places from which a part might be trying to capture its next type machine needed. Contained within each such row are the numbers of locations occupied by the next type of machine needed, arranged in the row from right to left in order of next closest location. (Transactions use a 'count down to zero' LOOP block to scan the appropriate row in right to left order.)

A matrix is also used to store the various machining step times (the STIME matrix) as a function of part type (row entry) and step number (column entry).

Each matrix in the model is populated by using the GETLIST statement in GPSS/H to read directly from the appropriate disk file into the corresponding matrix. The disk files themselves contain blank-delimited values and are easily populated using an operating system editor. This GETLIST approach is vastly superior to using tediously keyed matrix INITIAL statements, or the C form of the HELP block, one or the other of which would have to be done using most other implementations of GPSS.

In concluding this section, recall that the temptation to make the model highly compact, at the probable expense of ease of readability, was deliberately avoided. As a result, the model contains more statements than it would otherwise, with some blocks of statements repeated almost verbatim in a number of places in the model. This approach supports ease of study of the model. Others who use this model as a kernel for GPSS modeling of FMSs may prefer modifying the approach to achieve a higher degree of compaction.

5. A glimpse of the GPSS/H model

Figure 5.1 provides a glimpse of the listing of the GPSS/H model for the fig. 3.1 FMS, showing the initial processing of type 1 parts in the system. This glimpse is included to convey some feeling for the character of the model, and for the extent to which explanatory comments are embedded within the model to enhance its readability. (Each entry in fig. 5.1 beginning with an asterisk provides either a comment, or an otherwise blank line used as a spacer.) A complete listing of the model and the output produced by performing a simulation with it can be obtained from the author upon request.

```
*
*______________________________________________________________________
*
*         Model Segment 1
*
*         (Logic for Type 1 Parts)
*
*______________________________________________________________________
*
*
*         Check into the Part Type 1 Load Station/Fixture/Pallet Queue
 TYPE1    QUEUE     PT1LSFPQ
*
*         Wait for the Part Type 1 Load Station,
*         a Part Type 1 Fixture, and a Pallet
          TEST E    BV$PT1LSFP,TRUE
*
*         Check out of this Load Station/Fixture/Pallet Queue
          DEPART    PT1LSFPQ
*
*         Claim the Part Type 1 Load Station
          SEIZE     LSPT1
*
*         Remember this Part's current location
          ASSIGN    MYLOC,LSPT1,PF
*
*         Claim a Part Type 1 Fixture
          ENTER     PT1FIX
*
*         Claim a Pallet
          ENTER     PALLETS
*
*         Adjust Part's Priority for later use of a Step 1 Machine
          PRIORITY  1000-MH$STIME(PTYPE1,STEP1)
*
*         Put the Part into the Fixture
          ADVANCE   FIXTIME
*
*         Check into the Queue for Part Type 1, Step 1
*         (waiting at the Loading Station)
          QUEUE     PT1S1LSQ
*
*         Wait for a Cart and a Type A Machine
          TEST E    BV$CARTANDA,TRUE
*
*         Check out of the Queue for Part Type 1, Step 1
*         (coming from the Loading Station)
          DEPART    PT1S1LSQ
*
*         Find and claim the nearest idle cart
          TRANSFER  SBR,GETCART,RETURN$PF
*
*         Find and claim the nearest idle Type A Machine
          TRANSFER  SBR,GETTYPEA,RETURN$PF
*
*         The Cart travels to the Load Station
          ADVANCE   MH$TTIME(PF$CLOC,PF$MYLOC)
*
```

Fig. 5.1. Listing of a portion of the GPSS/H model.

6. Comments on output from the model

Figure 6.1 shows a portion of the output produced by the FMS model at the end of a 25-shift (200 hour) steady-state simulation. (There is no work in process in the model initially. Estimated steady-state conditions were brought about by simulating for 5 8-hour shifts, reinitializing various statistical accumulators to eliminate the transient response, and then resuming the simulation.)

```
          --AVG-UTIL-DURING--
FACILITY  TOTAL  AVAIL  UNAVL        ENTRIES     AVERAGE
           TIME   TIME   TIME                   TIME/XACT
MTYPEA1    .997                           60     199.433
MTYPEB2    .690                           65     127.446
WSPACE1    .267                           55      58.345
MTYPEC1    .656                           88      89.545
```

(a) Selected facility statistics

```
          --AVG-UTIL-DURING--
 STORAGE  TOTAL  AVAIL  UNAVL        ENTRIES     AVERAGE
           TIME   TIME   TIME                   TIME/UNIT
   CARTS   .201                          673      10.777
  MTYPEA   .998                          173     207.797
  WSPACE   .083                          109      55.192
  MTYPEB   .712                          138     123.934
  MTYPEC   .656                           88      89.545
  PT1FIX   .788                           91     519.791
```

(b) Selected storage statistics

```
TABLE  PT2RATE

ENTRIES IN TABLE   MEAN ARGUMENT   STANDARD DEVIATION
              25          2.0000                 6454

      UPPER     OBSERVED        PERCENT  CUMULATIVE  CUMULATIVE
      LIMIT    FREQUENCY       OF TOTAL  PERCENTAGE   REMAINDER
          1            5          20.00       20.00       80.00
          2           15          60.00       80.00       20.00
          3            5          20.00      100.00        0.00
```

(c) Production rate table for part type 2

Fig. 6.1. Sample output from the FMS model (based on simulations of 25 8-hour shifts).

Part (a) in fig. 6.1 shows selected statistics for the facilities used to model the type A machine at point (4) in fig. 3.1, the type B machine at point (6), the wait space at point (7), and they type C machine at point (14). These facilities are respectively given the symbolic names MTYPEA1, MTYPEB2, WSPACE1, and MTYPEC1 in the model, and the output is correspondingly labeled. Referring to the MTYPEC1 row in part (a) of fig. 6.1, for example, we see that this type C machine was in use 65.6 percent of the time (AVG UTIL DURING TOTAL TIME), was used on 88 occasions (ENTRIES), and had an average usage time of 89.545 min (AVERAGE TIME/XACT).

Part (b) in fig. 6.1 shows selected statistics for the storages used to model the pool of AGVs, the pools of type A, type B, and type C machines, the pool of wait spaces, and the pool of fixtures for parts of type 1. The respective model names and output labels for these resource pools are CARTS, MTYPEA, MTYPEB, MTYPEC, WSPACE, and PT1FIX. The CARTS row in part (b) of fig. 6.1 shows that the AGVs experienced an average use of 20.1 percent (AVG UTIL DURING TOTAL TIME), were used on 673 occasions (ENTRIES), and experienced an average time per use of 10.777 min (AVERAGE TIME/UNIT).

Part (c) of fig. 6.1 shows the production-rate table for type 2 parts, based on 8-hour shifts. Twenty-five observations were taken on this random variable (ENTRIES IN TABLE). The mean 8-hour production rate for type 2 parts was 2 (MEAN ARGUMENT), the standard deviation (biased if autocorrelated) is 0.6454 (STANDARD DEVIATION), and the relative frequencies of producing 1, 2, and 3 (UPPER LIMIT) type 2 parts in an 8-hour shift were 20, 60, and 20 percent, respectively (PERCENT OF TOTAL).

The following items (some of which have been described above) are included as part of the overall information provided in the standard fixed form, fixed content output produced after the 200-hour steady-state simulation.

(1) Production rate tables for type 1, type 2, and type 3 parts.

(2) The fraction of time in use, capture count, and average holding time per capture, of each individual machine and each individual wait space.

(3) The fraction of time in use, maximum number in use simultaneously, capture count, and average holding time per capture, of each pool of resources (carts; type A, type B and type C machines; wait spaces; pallets; and any part type 1, 2, and 3 fixtures).

(4) For each type of part waiting for a cart and its first type of needed machine (so that it can leave its loading station), the average wait time and the number of occasions when no waiting was necessary.

(5) For each type of part occupying a machine and waiting for a cart and either its next type of needed machine or a wait space (so that it can leave the machine), the average number experiencing such waiting, and the number of occasions when no waiting was necessary. This information is available as a function of the type of next machine needed.

(6) For each type of part occupying a wait space (waiting for a cart and its next type of needed machine), the average wait time, the average number experiencing such waiting, and the number of occasions when no waiting was necessary. This information is available as a function of the type of next machine needed.

Space limitations make it impossible to show additional output from the GPSS/H FMS model here. A complete set of output from a simulation performed with the model will be provided with a listing of the model upon request.

7. Potential uses for the model

For the FMS as configured in fig. 3.1, the GPSS model could be used to investigate the influence on production rate and resource utilization of such variables as the number of AGVs in the system, the number of pallets, the number of various types of fixtures, the total level of in-process inventory permitted, the criteria used to dispatch waiting parts to machines, and the criteria used to determine the order in which new part types are admitted to the system.

Furthermore, the fig. 3.1 configuration itself could be modified to reflect alternative locations for the machine and wait-space resources, then the correspondingly modified model could be used to investigate the resulting influences on production rate and resource utilization, perhaps as a function of alternative settings for the variables cited in the preceding paragraph.

In addition, the number of fixed-location resources (machines, and wait spaces) in the FMS could be varied and the resulting influence on production rate and resource utilization could be studied, perhaps as a function of the locations chosen for these resources, and again perhaps as a function of alternative settings for the variables cited above.

In summary, an FMS is characterized by many design and control variables. The influence of these variables on FMS performance can be examined through simulation modeling to answer such questions as: Which variables are important? What is their relative degree of importance? What are the most effective guidelines to use in controlling an FMS? And how can the performance required of an FMS be achieved at least cost?

8. Applying the GPSS/H model to other FMSs

The GPSS/H model for the fig. 3.1 FMS has been built in a highly modular fashion. In effect, the model segment used by a part to migrate from its loading station to its first machine is a module; the model segment used by a part to migrate from its jth to its (j + 1)st machine, or to a wait space and then to its (j + 1)st machine, is a module; and the model segment used by a part to migrate from its last machine to its unloading station is a module. Given the existence of each of these three fundamental modules, use of an operating system editor to copy selected modules into appropriate positions in the model file and then interface them and tailor them, making them specific to the type of part and type of machine being represented at those positions, is straightforward and can be done quickly. The GPSS/H model presented in this work was itself built in this bootstrap fashion. (An alternative approach would be to define each module as a GPSS macro, then specify the appropriate macro and macro arguments at various points in the model.) Furthermore, the fundamental modules used in this model can be taken as starting points by others who need to model FMSs structured in ways other than that shown in fig. 3.1.

Using GPSS/H as a medium, and taking the GPSS/H model for fig. 3.1 (or the types of modules described above) as a starting point, the notion of modeling more sophisticated FMSs can also be easily entertained. Some or all of the simplifying assumptions described at the end of sect. 3 can be relaxed, as well as other implicit assumptions not mentioned there. Incorporating random machine breakdowns of random duration into the model would be relatively straightforward, for instance. And so it should be quite reasonable to apply aspects of this model to other FMSs.

9. Summary

This paper describes some of the pertinent characteristics and application potential of a GPSS/H model built for a hypothetical flexible manufacturing system. The model has been built in a modular fashion and has been structured and documented in a way designed to ease the process of coming to an operational understanding of it. Those who reach such understanding should be in a position to apply and extend the techniques demonstrated in the model to other FMSs which are of immediate interest to them.

Acknowledgements

The benefits of fruitful discussion of flexible manufacturing systems with Professors Kathryn E. Stecke and F. Brian Talbot of the University of Michigan, and the valuable insights of two referees, are acknowledged with thanks.

References

[1] D.P. Christy and H.J. Watson, The application of simulation: A survey of industry practice, Interfaces 5 (Oct. 1983).
[2] S. Cox, *GPSS/PC* (MINUTEMAN Software, Stow, MA, 1984).
[3] J.O. Henriksen, An improved events list algorithm, *Proc. 1977 Winter Simulation Conf.* (Society for Computer Simulation, San Diego, CA, 1977).
[4] J.O. Henriksen, GPSS: Finding the appropriate world view, *Proc. 1981 Winter Simulation Conf.* (Society for Computer Simulation, San Diego, CA, 1981).
[5] J.O. Henriksen and R.C. Crain, *GPSS/H User's Manual,* 2nd edition (Wolverine Software Corporation, Annandale, VA, 1983).
[6] D. Martin, *GPSS/VAX Reference Manual* (Simulation Software Design and Development Company, London, Ontario, Canada, 1981).
[7] D.L. Martin and K.J. Musselman, Simulation in the life cycle of flexible manufacturing systems, *Proc. 1st ORSA/TIMS Special Interest Conf. on Flexible Manufacturing Systems* (The Institute of Management Sciences, Providence, RI, 1984).
[8] T.J. Schriber, *Simulation Using GPSS* (Wiley, New York, 1974).
[9] K.E. Stecke, Design, planning, scheduling, and control problems of flexible manufacturing systems, this volume, Annals of Oper. Res. 3(1985)3.