PERSPECTIVES ON SIMULATION USING GPSS

Thomas J. Schriber

Computer and Information Systems
Graduate School of Business Administration
The University of Michigan
Ann Arbor, Michigan 48109-1234

ABSTRACT

A brief perspective on GPSS (General Purpose Simulation System) is presented. The approach taken in GPSS to model a one-line, one-server system is explained, implementation details are provided, and results are discussed. Suggestions for further study are given.

1 GPSS IN BRIEF

GPSS (General Purpose Simulation System) is a simulation modeling language used to build computer models for discrete-event simulations. (A discrete-event simulation is one in which the state of the system being simulated changes at only a discrete set of time points.) GPSS lends itself especially well to modeling systems in which discrete units of traffic compete with each other for the use of scarce resources, and is useful in determining how well such systems will respond to the demands placed on them. GPSS has been applied, for example, to the modeling of manufacturing systems, communication systems, computing systems, transportation systems, inventory systems and health-care systems, among others.

2 GPSS SEMANTICS AND SYNTAX

GPSS offers rich semantics with sparse syntax. For example, only seven statements (plus several run-control statements) are required to model a one-line, one-server queuing system in GPSS. These statements take such simple forms as "GENERATE 18,6" and "QUEUE LINE". No read, write, format, or test statements appear in the model. And yet, when a simulation is performed with the model, fixed-form, fixed-content output is produced, providing statistics for the server (e.g., number of times captured; average holding time per capture; fraction of time in use) and the waiting line (e.g., average contents; maximum contents; average time in line), etc. This limited example is roughly suggestive of the character of GPSS. A GPSS model for a one-line, one-server system is given here in Appendix A.

The sparse syntax of GPSS, coupled with its block-diagram orientation, makes it possible for the beginner to learn quickly a usable subset of the language. Because GPSS is rich and versatile, however, serious study is required to *master* the language.

3 VARIOUS GPSS IMPLEMENTATIONS

GPSS is a multi-vendor language. (This is in contrast with such languages as SIMAN, SLAM, and SIMSCRIPT.) Brief comments are provided here on several GPSS vendors and their GPSS implementations.

Wolverine Software Corporation (4115 Annandale Road, Annandale VA 22003-2500, phone 800-456-5671 or 703-750-3910) vends GPSS/H, Release 2, which runs on a wide range of hardware platforms from DOS machines through engineering workstations to mainframes. In addition to the vendor's GPSS/H reference manual [Henriksen and Crain 1989], two textbooks describe this implementation in introductory fashion [Banks, Carson and Sy 1989, and Schriber 1991] and come with Student DOS GPSS/H on an included disk. The vendor sponsors GPSS/H courses and courses teaching Proof, the vendor's high quality and inexpensive presentation and animation software.

MINUTEMAN Software (P.O. Box 171, Stow MA 01775-0171, phone 800-223-1430 or 508-897-5662) vends a series of GPSS/PC implementations. GPSS/PC provides built-in graphics and animation and in some versions interfaces with AUTOCAD. The vendor supplies a reference manual [Cox 1988] and a series of tutorial models [Cummings 1988]. A textbook containing Student GPSS/PC on an included disk is Karian and Dudewicz [1991]. The vendor has information about courses that train participants in the use of GPSS/PC. Contact the vendor for information about courses, books, and reference materials.

Simulation Software Ltd. (760 Headley Drive, London Ontario Canada N6H 3V8, phone 519-657-8229) vends a series of GPSS implementations. Contact the vendor for details.

International Business Machines (IBM) leases GPSS V, its early 1970's implementation of GPSS which, although out of date, is still used occasionally. Contact your IBM representative for details.

4   THE GPSS TUTORIAL

In the GPSS tutorial at the Winter Simulation Conference, the fundamentals of queuing system logic and the elements offered by GPSS to implement this logic will be discussed. A GPSS model for a one-line, one-server queuing system is given below in Appendix A for interested persons unable to attend the tutorial.

APPENDIX A: A GPSS MODEL FOR A ONE-LINE, ONE-SERVER SYSTEM

This appendix presents a GPSS model for a one-line, one-server queuing system. Although the model is largely generic to GPSS, its implementation is shown in GPSS/H. Animation of the model is not discussed but could be accomplished in MINUTEMAN Software's GPSS/PC or by using Wolverine Software's animation product, Proof. Contact the vendors for details.

The appendix consists of these sections:

A.1   Statement of the Problem
A.2   The Approach Taken in Building the Model
A.3   The GPSS Block Diagram for the Model
A.4   The GPSS Model File
A.5   Selected Simulation Output
A.6   Suggestions for Further Study

A.1   Statement of the Problem

In a manufacturing system, castings are sent to a drill, where each casting is to have a hole drilled in it. The interarrival time of castings at the drill is uniformly distributed over the interval 15.0 ± 4.5 minutes. The time required to drill a hole in a casting is 13.5 ± 3.0 minutes, uniformly distributed. Castings are processed in first-come, first-served order. Model this system in GPSS, making provision to collect queuing statistics for castings waiting their turn to be drilled. Perform a single simulation with the model, simulating until holes have been drilled in 100 castings. Briefly discuss the output produced at the end of the simulation.

A.2   Approach Taken in Building the Model

Consider the series of events experienced by a casting as it moves through the one-line, one-server system:
1. The casting arrives at the system.
2. The casting requests the drill.
3. The casting waits, if necessary, to capture the drill.
4. The casting captures the drill.
5. The casting holds the drill in a state of capture while a hole is drilled in the casting.
6. The casting gives up control of the drill.
7. The casting leaves the system.

Castings can be thought of as units of traffic (objects) that move through the castings-and-drill system. These units of traffic are conveniently simulated in GPSS by language elements known as "transactions."

Transactions are units of traffic which are created and introduced into a model from time to time, move along a path in the model as the simulation proceeds, and then are destroyed (leave the model). The experiences of transactions as they go through their life cycle in the castings-and-drill *model* are analogous to the experiences of castings as they go through the castings-and-drill *system*. Positioned on the path along which transactions move are *blocks*. Movement of a transaction into a block causes the block to be executed. By choosing appropriate types of blocks, the GPSS modeler can easily build an appropriate path (sequence of blocks) for casting-transactions to move along to mimic the sequence of events outlined above.

The sequence of blocks begins with the type of block used to create transactions from time to time during a simulation and introduce them into a model, the GENERATE block. The time that elapses between introduction of consecutive transactions into a model by a GENERATE block is "interarrival time." In this model, the interarrival time is uniformly distributed over the interval 15.0 ± 4.5 minutes. (15.0 ± 4.5 describes the open interval ranging from 10.5 to 19.5.) The values 15.0 and 4.5 are provided in the model as GENERATE block *operands*. (In general, arbitrary interarrival-time distributions can be modeled at GENERATE blocks. This is done by using built-in or user-defined functions that describe the distribution, then specifying these functions as GENERATE-block operands. See Schriber [1991], chapter 13, for particulars.)

The sequence of blocks ends with a TERMINATE block. When a transaction executes a TERMINATE block, the block destroys the transaction. A counter can be used with TERMINATE blocks so that, after a specified destroy count has been reached (a count of 100 in this problem), a simulation will stop. (More generally, arbitrarily complicated stopping conditions can be specified in GPSS models.)

A SEIZE block is included in the sequence. A transaction requests control of a single server by trying to execute a SEIZE block. A SEIZE block operand is used to identify the single server. If the server is idle when a transaction requests it, the requesting transaction executes the SEIZE without delay and takes control of the server. But if the server is currently under the control of one transaction when another requests it, the requesting transaction cannot execute the SEIZE block. Instead, it remains in its current block and waits its turn to capture the server. In the simplest case, turns come in the order of first-come, first-served. (In general, arbitrarily complicated service orders can be specified in GPSS.)

A RELEASE block is also included in the sequence. A transaction which is in control of a single server gives up control by executing a RELEASE block. A RELEASE block operand is used to identify the server involved.

GPSS automatically collects (and then, when a simulation stops, prints out) statistical information about

single servers modeled with use of SEIZE and RELEASE blocks. (See section A.5.)

An ADVANCE block is used to delay movement of a transaction along its path for a specified simulated time. In this model, an ADVANCE block can be used to simulate the time required for the machine to drill a hole in a casting ("service time"). The service time in this model is uniformly distributed over the open interval 13.5 ± 3.0 simulated minutes. The values 13.5 and 3.0 are provided in the model as ADVANCE block operands. (Arbitrarily complicated service time distributions can be modeled at ADVANCE blocks. This is done by using built-in or user-defined functions which describe the applicable distribution.) By placing an ADVANCE on the path between SEIZE and RELEASE, simulated time delays between server capture and release can be modeled.

By executing a QUEUE block, a transaction initiates membership for itself in a *queue*, or waiting line. This membership continues until the transaction brings its queue membership to an end by executing a DEPART block. An operand is used at the QUEUE and DEPART blocks to indicate the particular queue involved. By placing a SEIZE between QUEUE and DEPART blocks, transactions will be members of a queue while waiting their turn to capture a server. GPSS automatically collects and then prints out statistical information about such queues. (See section A.5.)

Seven types of blocks have been commented on in this section (GENERATE; TERMINATE; SEIZE; RELEASE; ADVANCE; QUEUE; DEPART). In total, there are more than *fifty* types of blocks in GPSS. By appropriate use of these block types, models of complex systems can be built with considerable ease.

A.3 The GPSS Block Diagram for the Model

The model described above is shown in the form of a block diagram in Figure A-1. The block diagram consists of a sequence of seven Blocks. (Each block type in Figure A-1 has its own unique, arbitrary geometry.)

The text appearing adjacent to the blocks in Figure A-1 (e.g., "castings arrive"; "check into the drill queue") is not part of the model, but is simply commentary which has been (optionally) provided as documentation.

A.4 The GPSS Model File

Figure A-1 shows the *block diagram* for a GPSS one-line, one-server model. To perform a simulation with this model, the *statement version* of the Figure A-1 block diagram must be prepared and then supplemented with additional types of statements used to control compilation and execution of GPSS models. The resulting collection of statements must then be arranged in a model file. A model file is simply a computer file which can be used as the basis for performing one (or more) simulations.

Figure A-2 shows a model file corresponding to the

Figure A-1 block diagram. The Figure A-2 model file has been supplemented for discussion purposes here at the top with a row of column labels ("STMT #," "Label," "Operation," "Operands," and "Comments") and at the left with a column of statement numbers (1, 2, 3, ..., 22).

Statements 8 through 14 correspond to the Blocks in Figure A-1. Each of these block statements consists potentially of a "Label" (no Labels are used in Figure A-2), an "Operation," and zero or more "Operands," and can (optionally) include appended documentation text ("Comments"). For example, STMT #8 corresponds to the Figure A-1 "GENERATE 15.0,4.5" block (where the "Operation" is GENERATE and the "Operands" are 15.0 and 4.5) and carries the optional appended comment "castings arrive."

Statements 2, 20 and 22 in Figure A-2 are examples of statements used to control the compilation and execution of GPSS models. They have been specified in Figure A-2 in such a way that when the model file is executed, only one simulation will take place. The simulation will stop when the 100th casting has been drilled.
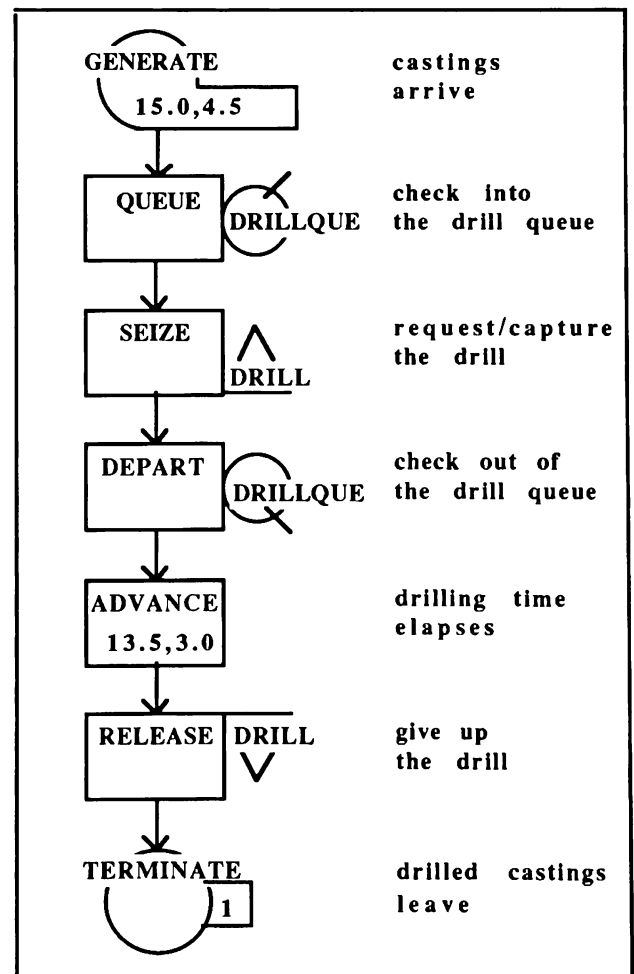

Figure A-1: GPSS Block Diagram for a One-Line, One-Server Queuing System

STMT #	Label Operation Operands Comments
1	*************** One-Line, One-Server Model, WSC '91 ***************
2	SIMULATE
3	* Base Time Unit: 1 Minute
4	**
5	* Model Segment 1 (Movement of Castings Through the System *
6	**
7	*
8	GENERATE 15.0,4.5 castings arrive
9	QUEUE DRILLQUE check into the drill queue
10	SEIZE DRILL request/capture the drill
11	DEPART DRILLQUE check out of the drill queue
12	ADVANCE 13.5,3.0 drilling time elapses
13	RELEASE DRILL give up the drill
14	TERMINATE 1 drilled casting leaves
15	*
16	**
17	* Run-Control Statements *
18	**
19	*
20	START 100 start the simulation; proceed until
21	* 100 drilled castings have left
22	END end of Model-File execution

Figure A-2: A GPSS Model File for the Figure A-1 Block Diagram

Each model-file statement beginning with an asterisk (*) is a *comments statement*. In Figure A-2, STMT #'s 2, 3 through 7, 15 through 19, and 21 are examples of such statements.

A.5 Selected Simulation Output

Selected output automatically produced at the end of the simulation when the Figure A-2 model file was submitted for execution is displayed in Figure A-3. The output in Figure A-3 consists of: (a) clock values; (b) server statistics; and (c) queue statistics. Portions of this output are discussed below. (For a full discussion of similar output, see Banks, Carson and Sy 1989, Henriksen and Crain 1989, or Schriber 1991.)

(a) Clock Values

As indicated in Figure A-3(a), GPSS maintains two simulated clocks: a RELATIVE CLOCK; and an ABSOLUTE CLOCK. Both clocks show that it took 1488.9+ simulated minutes to drill holes in 100 castings. (Limited space makes it impossible to explain the difference between the two types of clocks here.

(b) Server Statistics

Figure A-3(c) shows server (drill) statistics accumulated during the simulation. Several columns in the figure have been numbered here to make it easy here to refer to

the information they contain. The meaning of the information in the several numbered columns will now be indicated by column number.

(1) The FACILITY column lists the identifier used in the model for the single server (the DRILL, in this case) for which statistics are being reported.

(In GPSS, the *facility* entity is used to model single servers.)

(2) The --AVG-UTIL-DURING-- TOTAL TIME column shows the *fraction* of *total simulated time* that the server was captured. In this case, the DRILL was in use 91.7% of the time.

(3) The ENTRIES column indicates the number of times the server was put into a state of capture during the simulation. This statistic is a *capture count*. In Figure A-3(c), the capture count is 100.

(4) The AVERAGE TIME/XACT column shows the average holding time per capture of the server.)

(c) Queue Statistics

Figure A-3(d) shows queue (waiting-line) statistics accumulated during the simulation. Several columns in the figure have been numbered here to make it easy here to refer to the information they contain. The meaning of the information in the several numbered columns will now be indicated by column number.

(1) The QUEUE column lists the identifier used in the model for the queue (the DRILLQUE, in this case) for which statistics are being reported.

RELATIVE CLOCK: 1488.9629	ABSOLUTE CLOCK: 1488.9629
(a) Clock Values	

(1) FACILITY	(2) --AVG-UTIL-DURING-- TOTAL TIME	AVAIL TIME	UNAVL TIME	(3) ENTRIES	(4) AVERAGE TIME/XACT	CURRENT STATUS	PERCENT AVAIL	SEIZING XACT
DRILL	.917			100	13.655	AVAIL	100.0	
(b) Drill Statistics								

(1) QUEUE	(2) MAXIMUM CONTENTS	(3) AVERAGE CONTENTS	(4) TOTAL ENTRIES	(5) ZERO ENTRIES	PERCENT ZEROS	(6) AVERAGE TIME/UNIT
DRILLQUE	2	.215	101	42	41.6	3.172
(c) Queue Statistics						

Figure A-3: Selected Simulation Output

(2) The MAXIMUM CONTENTS column indicates the maximum length of the waiting line (this statistic has the value 2 in the case of the DRILLQUE).

(3) The AVERAGE CONTENTS column shows the average length of the waiting line (0.215 in the case of the DRILLQUE).

(4) The TOTAL ENTRIES column shows the count of the number of times transactions joined the waiting line (101 in the case of the DRILLQUE).

(5) The ZERO ENTRIES column shows the number of transactions which passed through the waiting line in zero simulated time.

(6) The AVERAGE TIME/UNIT column shows how much time transactions spent resident in the waiting line on average (3.172 in the case of the DRILLQUE). (Here, the term "UNIT" in the AVERAGE TIME/UNIT label means "transaction.")

A.6 Suggestions for Further Study

The preceding material provides a glimpse at the particulars of discrete-event simulation using GPSS. Those interested in further exploration can do the following:

1. Contact the vendors and obtain specific information about the various implementations.

2. When attending a conference (such as the Winter Simulation Conference) at which simulation vendors have booths in the exhibition area, talk with the vendors and look at vendor demonstrations.

3. Obtain and read a textbook for the implementation(s) of interest and, if the textbook(s) come with student software and sample models on an included disk, experiment directly with the software.

4. Read GPSS application papers in areas of interest.

5. Take an intensive GPSS short course.

REFERENCES

Banks, J., J.S. Carson, and J. Sy. 1989. *Getting Started with GPSS/H* (with Student DOS GPSS/H on an included disk). Wolverine Software, Annandale, VA.

Bobillier, P.A., B.C. Kahan, and A.R. Probst. 1976. *Simulation with GPSS and GPSS/V*. Prentice-Hall, Englewood Cliffs, NJ.

Cox, S. 1988. *GPSS/PC User's Manual*. MINUTEMAN Software, Stow, MA.

Cummings, G.F. 1988. *GPSS/PC Simulation Tutorials*. MINUTEMAN Software, Stow, MA.

Gordon, G. 1975. *The Application of GPSS V to Discrete Systems Simulation*. Prentice-Hall, Englewood Cliffs, NJ.

Henriksen, J.O. and R.C. Crain. 1989. *GPSS/H Reference Manual*, Third Edition. Wolverine Software Corporation, Annandale, VA.

International Business Machines. 1970. *GPSS V User's Manual* (SH20-0851). IBM Inc., Armonk, NY.

Karian, Z.A., and E.J. Dudewicz. 1991. *Modern Statistical, Systems GPSS Simulation: The First Course*. W.H. Freeman, New York.

Schriber, T.J. 1974. *Simulation Using GPSS*. John Wiley & Sons, New York. (As of 1990, this book is published by Krieger Publishing, Melbourne FL.)

Schriber, T.J. 1991. *An Introduction to Simulation Using GPSS/H* (with Student DOS GPSS/H on an included disk). John Wiley & Sons, New York.

AUTHOR BIOGRAPHY

THOMAS J. SCHRIBER is Professor and Chairman of Computer and Information Systems in the Graduate School of Business at The University of Michigan. He teaches, does research, and consults in the area of discrete-event simulation. A member of Who's Who in America, he has authored or co-authored several dozen articles, has authored or edited eleven books, including *An Introduction to Simulation Using GPSS/H* (Wiley, 1991), and regularly teaches intensive courses on GPSS-based simulation. From 1977 to 1986 he was the ACM member of the Board of Directors of the Winter Simulation Conferences, serving as Board Chairman two years. His professional affiliations include ACM, DSI, ORSA, SCS, and TIMS.