General Purpose Simulator

TATSUHEI UCHIYAMA*, EIJI HORIKAWA*, SEISUKE SVGAWARA*,
TOSHIHIKO NISHIKI*, YUKIO MIZUNO**, MASANOBU KIMURA**,
AND KENICHI MATSUKI**.

1. *Introduction*

Although Computer Simulation usually requires a large scale computer, the person who wants to do simulation, does not always have a large scale one. In consideration of this problem, we have developed a simulation program compiler for a small scale computer of the NEAC-2200 series. As the result of surveying various kinds of simulation language, we decided to make use of the flow-oriented language of GPSS-II [2]. We have adopted the format of GPSS-II and added some functions in which GPSS-II is lacking. To make up for the functions which GPSS-II lacks, we have added some convenient functions. An external magnetic drum access file has also been added to overcome the small capacity of core memory. Random Numbers (abbreviated in the following as RN) necessary for simulation are generated by the Random Number Generator Equipment. (RNGE) This system program is termed FOMOS (Flow Oriented Monte Carlo Simulator).

2. *Random Number Generator Equipment*

It is necessary for simulation by the Monte Carlo method to generate unbiasedly and uniformly a great many random numbers at high speed. RN is usually created by means of pseuo random number method, ie. Lehmer's method. But the nature of pseudo RN is cyclic and also has a tendency to become biased. As RN generated by hardware equipment has no cycles, we can use RN without cycles. We have succeeded in developing a RNGE which is connected directly with the computer. Move details of RNGE will be explained. This system obtains RN by using the 'Noise of the Thyratron.' The noise generated in a noise source of the thyratron is clipped at certain voltages and a pulse is generated at each crossing point (NG1-NG7) and these noises trigger flip-flops.

The pulse of OR gate (G1) and the pulse of the thyratron (NG1) are composed at AND gate (G11), and this noise triggers the flip-flops. When the signal surpas-

This paper first appeared in Japanse in Joho Shori (the Journal of the Information Processing Society of of Japan), vol. 8, No. 2 (1967), pp. 73-80

* Industrial Engineering Department, Yawata Works, Yawata Iron & Steel Co., Ltd.
** Software Development Department, Data Processing Division, Nippon Electric Co., Ltd.

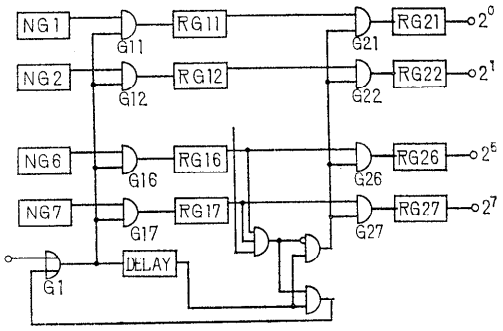

Fig .1.

ses a predetermind constant level, the stage of flip-flop register (RG11) is changed from "zero" state to "one" state. As this RN is on a binary system, uniformly-distributed-RN are obtained by means of a combination of registers (RG11–RG17). If seven registers are combined, we get a pure binary value (2^0–2^7), i. e. 0–127. These binary values are inconvenient for use in the decimal system, but we can get '0–99' value by using a certain combination of gates.

The uniormity of the random numbers is checked by both the hardware checker and the software programs. The expected frequency of each register for the "zero" state or the "one" state will be 50%. After every 1024 pulses, a hardware error indicator is automatically set if l's state check register contains a value not within ±3% of the expected times. When we get two RN scquencially from the same register, the combination of two bits are specified as follws; '00, 01, 10, 11'. Each expectation of combination is one-fourth. The expected value of frequency is also tested by the hardware checker. As the software check, "Chi-square test' is used for the uniformity of RN. The theoretical analysis of uniformity is done by [1]. The stability of this equipment is carefully ascertained. This system program is used, as stated earlier, in a NEAC-2200 series computer. This is a 2 address and variable length computer. One character consists of 6 data bits and 2 punctuation bits. RN data are read in through PDT instruction, as in Fig. 2.

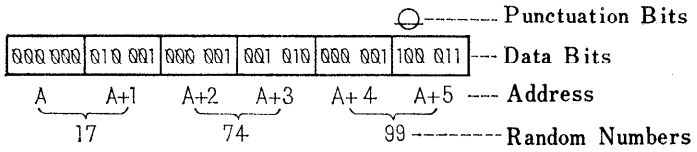

Fig. 2.

One RN datum needs 2 characters and they are read into the core memory up to the character with punctuation mark bits. Its value is a pure binary from 00

to 99. (Only one PDT instruction gets RN as the occasion demands). This RN is used also as the argument for functions.

3. *Hardware Configuration*

The hardware configuration of this system is as follows. As this system has been considered as simulation oriented, a magnetic drum is employed for the system program and data stored file. This random access file is superior to the magnetic tape unit, as far as the simulation speed is concerned.

Input Media :	Paper Tape Reader or Card Reader Random Number Generator
Output Media :	Console Typewriter or Line Printer Penwriting oscillator
Main :	20K chrs. and Megnetic Drum for External Auxiliary Memory

The penwriting oscillator is used for visualization of data being sampled out during the simulation run.

4. *Language*

Based on experiences with many types of simulations [3] and on analysis of of GPSS-II language, we decided to adopt a set of 39 blocks. A block diagram is used to model the structure of the system. The following functions are not included in GPSS-II but we have included them in FOMOS for additional capabilities:

a) Expansion of QUEUE

The unit upon which the system operates is called a "Transaction." In GPSS-II a block called QUEUE is used to cxpress the waiting conditions of various units. The QUEUE block alone does not permit making decisions in accordance with the conditions (e. g. the waiting times, length of QUEUE, etc.) of a transaction in waiting. With this in mind, FOMOS language has two new blocks (WAIT and DEPART) which replace the former QUEUE block.

b) High hLevel Use of Storage

Two new blocks termed OBSTRUCT and CANCEL are added to the FOMOS language. These blocks can cause interruption for the transaction which use the storage equipment at the low level blocks-ENTER, LEAVE.

c) 'BYE' Selection Mode

While a transaction occupies an equipment, another trans with a higher priority can cause an interruption and use the equipment formerly occupied by the original trans. The transaction so interrupted goes into a waiting state. In actual simulation, however, it may occur that the interrupted transaction proceeds to go under the control of another operation. The FOMOS simulator has the capability of to

allowing the interrupted transaction to proceed to another route. This function is called the 'BYE' selection mode.

d) Sampling System

With the present GPSS-II, it is impossible to know the internal process of a simulation run. In order to overcome this difficulty, the FOMOS langauge samples the state of the simulation at given time intervals and makes available the imformation so obtained using a lineprinter or pen recorder. The function helps one follow and observe the course of simulation for any necessary analysis.

e) Expand of ASSEMBLE BLOCK

The ASSEMBLE block can assemble only the "same assembly set" transaction. It is also possible to assemble the "not same assembly set" transaction.

5. *Program*

The FOMOS systems is composed of three phase programs-input phase, scanning phase and output phase. The input phase reads simulation model cards and edits,

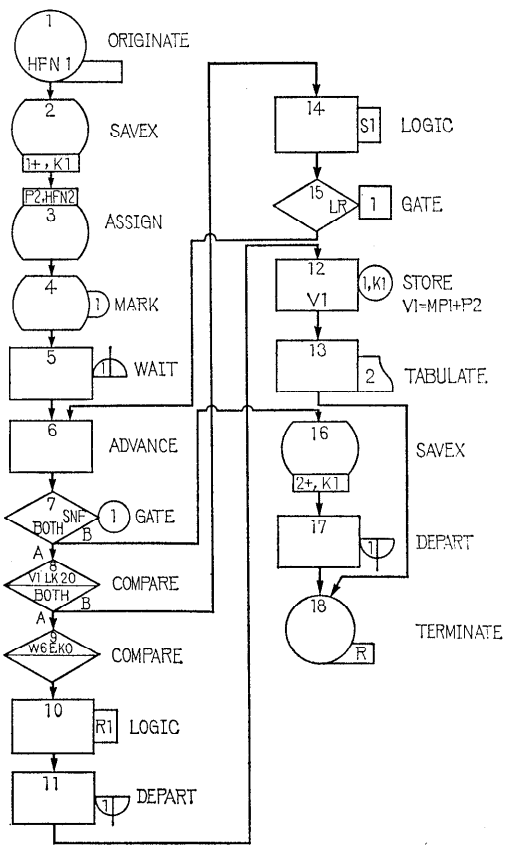

Fig. 3. The block chart of the process of heating the iron ingot.

compiles and checks them. If there is no error, main progam of scanning phase starts the the simulation scan and lists up all transaction that can move with this "block departure time" from the future event chain to the current event chain. Sampling dummy transaction are included in this transaction list and it is attempted to move transaction into some further block, and if successful, the control routine executes the block type type subroutine and calculates the next move transaction time. If unsuccessful, the test status is changed and control is transferred to the next transaction. After all transactions included in list are checked, the timing routine goes to next clock event. When a simulation CLOCK arrives at an assigned simulation end time, control goes to the output phase. The output phase prints out the result of the simulation run.

6. *Example*

We describe an example which is a simulation of iron ingots arriving at some furnaces to be reheated at a constant temperature before being passed on to some other process. It is assumed that the arrival rate of the ingots conforms to some probability distribution. A waiting line of iron ingots is made in front of the furnaces in a single queue.

Ordinarily the order of heating the iron ingots follows the principle of 'first in first out'. However, when there are two or more ingots in the queue, the processing order follows the principle of 'last in first out'. The required time of heating an iron ingot in a furnace is calculated as follows.

$$Th = Tw + Ci$$

where,

Th=heating time in the furnace
Tw=waiting time in front of the furnance
Ci=constant time defined by the classification of the iron ingot
(i=1, 2, ..., n)

In order to increase the efficiency of the furnaces, an iron ingot will not be accepted into a furnaces if its required heating time is in the excess of some given time unit (20 units in this paticular example). Each transaction interval (i. e. the arrival interval for an ingot) is generated by HFUNCTION 1. Heating time is represented by the 'VARIABLE 1'. It also determines whether or not the ingot will be heated in a furnace.

VARIABLE 1 is also used to determine the action time in a furnace. The above mentioned simulation model is shown in fig. 3.

Acknowledgement

Dr. S. Ura and Mr. K. Ohse of Keio Univ. provided valuable assistance in the development of the system.

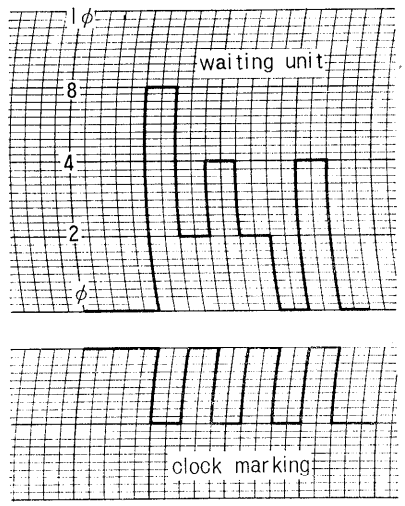

Fig. 4 Penrecorder output (example).

STORAGE

NR	CAPACITY	NUMBER ENTRIES (TOTL)	NUMBER ENTRIES (OBST)	AVERAGE CONTENTS (TOTL)	AVERAGE CONTENTS (OBST)	AVERAGE UTILIZ (TOTL)	AVERAGE UTILIZ (OBST)	AVERAGE TIME/UNITS (TOTL)	AVERAGE TIME/UNITS (OBST)	OBST/ (TIME)
1	2	836	0	1.582	0.000	0.7914	0.0000	11.918		0.000

QUEUE

NR	TABLE NUMBER	CONTENTS (MAXIMUM)	CONTENTS (AVERAGE)	ENTRIES (TOTL)	ENTRIES (ZERO)	AVERAGE TIME/UNITS	ZEROS/TOTAL ENTRIES	CURRE CONTE
1	3	7	0.614	1000	425	3.867	0.4249	

TABLE NUMBER 2

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
836	13.837	5.259	NON-WEIGHTED
836	13.837	5.259	WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	RELATIVE FREQUENCY	CUMULATIVE FREQUENCY	UPPER LIMIT /MEAN ARGUMENT	DEVIATION FROM MEAN
10	425	0.5083	0.508366	0.722	-0.72
12	85	0.1016	0.610043	0.867	-0.34
14	80	0.0956	0.705731	1.011	0.03
16	49	0.0586	0.764345	1.156	0.41
18	46	0.0550	0.819374	1.300	0.79
20	33	0.0394	0.858839	1.445	1.17
22	35	0.0418	0.900709	1.589	1.55
24	34	0.0406	0.941380	1.734	1.93
26	26	0.0310	0.972480	1.878	2.31
28	23	0.0275	0.999985	2.023	2.69
30	0	0.0000	0.999985	2.168	3.07
32	0	0.0000	0.999985	2.312	3.45
OVERFLOW	0	0.0000	0.999985		

Fig. 5 Simulation result (part of output).

References

[1] Sato. T. On The Stastical Properties of High Speed Binary Random Numbers Generated from Noise. *The Tokyo Institute of Technology 50*, (1962).

[2] General Purpose Simulator II. *IBM Reference Manual, B-206346.*

[3] Matsuyama, Y., S. Sugawara and T. Asagami, Computer Simulation of the Tobata Iron Ore Port Installation. *Preprint for ITEMS & ORSJ.* Joint Int'l Meeting J2. Tokyo (Aug., 1963).

[4] Kunisawa, K., H. Morimura and T. Mizuno, Queuing Simulator and Its Applications. *Preprint for TIMS & ORSJ.* Joint Int'l Meeting, C2, Tokyo, (Aug., 1963.)