

シミュレーション言語 : SOLとGPSS/60

武谷, 峻一
九州大学大学院工学研究科通信工学専攻

<https://doi.org/10.15017/1467974>

出版情報 : 九州大学大型計算機センター広報. 4 (1), pp.15-29, 1971-02-25. 九州大学大型計算機センター
バージョン :
権利関係 :

シミュレーション言語 — SOLとGPSS/60 —

武 谷 峻 一*

1. はじめに

このたび、富士通よりシミュレーション言語としてFACOM230-60 SOLとFACOM230-60 GPSS/60が提供された。SOLはセンターでのテストも終って一般利用できるようになり、GPSS/60は目下テスト中で近々利用できるようになる予定である。そこで、このSOLとGPSS/60とを簡単なプログラム例を示しながら紹介することにする。

2. コンピュータによるシミュレーション

我々が種々の問題を数学的に処理する場合、まず

- (1) 問題を数式化する。
- (2) 数式化された問題を解く。

ことが基本的な方法である。しかし、現実の問題として数式化できないものが数多くあり、また、たとえ数式化できても解析的に解くことが不可能な場合もある。とはいっても種々の問題に直面すれば、なんとか答を求めなければならない。こうした時にシミュレーションの手法が有効となる。

シミュレーションは一口でいえば“モデルによる実験”である。モデルをいろいろに設定した条件の下で動かすことにより、システムの構成要素間の相互関係をダイナミックにとらえることである。シミュレーションの特徴としては

- (1) 現象に含まれる諸因子、諸条件を忠実にモデルの中に含める。
- (2) 数式を作るといふより論理演算が多い。
- (3) 数式を解析的に解く必要がない。

などがあげられる。

さて、コンピュータでシミュレーションを行なう場合には、計算を主体としたFORTRANやALGOLでは記述しにくい部分が多い。すなわち、本来逐次的な計算手順を記述するために開発されたFORTRANやALGOLで、シミュレーションのように種々の事象が時間とともに並列に進行するものを記述するには高度のプログラミング技術が必要となる。このためシミュレーションを目的とした各種の言語が開発され、現在ではこのシミュレーション言語でモデルを記述するのが一般的である。シミュレーション言語は特殊なものを除き、一般に連続型モデルと離散型モデルとに大別できる。前者は変数が時間と共に連続的に変化するようなモデルであり、後者は離散的に変化するものである。

* 九州大学大学院工学研究科通信工学専攻

従来アナログコンピュータで行なっていたような微分方程式で表わされるモデルが前者の例であり、待ち行列モデルなどが後者の例である。著名なシミュレーション言語を分類してみると

○ 連続型シミュレーション言語

- CSMP (Continuous System Modeling Program)
- ADSL (Analogue to Digital Simulation Language)
- PACTLOUS, MIDAS

○ 離散型シミュレーション言語

- GPSS (General Purpose System Simulator)
- SOL (Simulation Oriented Language)
- GASP (General Activity Simulation Program)
- SIMSCRIPT (SIMulation SCRIPTor)

などがあり、この他にDYNAMO(DYNAmic MOdels)のように中間的なものもある。

このような言語を利用してシミュレーションを行なうには、一般に次に示すような手順を踏む(図1参照)。

- (1) ある問題が提示され、シミュレーションにより解析しようとする。
- (2) データを収集し、整理する。
- (3) モデルを作成する。モデルを一気に完成させることはむずかしく、徐々に複雑なモデルへと進展させた方がよい。
- (4) プログラミングを行なう。シミュレーション言語は一般にデバッグが非常に困難なので、はじめからデバッグを考慮してプログラムを作製した方が得策である。
- (5) プログラムを実行する。
- (6) 得られた結果を検討する。必要があればデータを変更し、場合によってはモデルに修正を加えてこの手順を繰り返す。

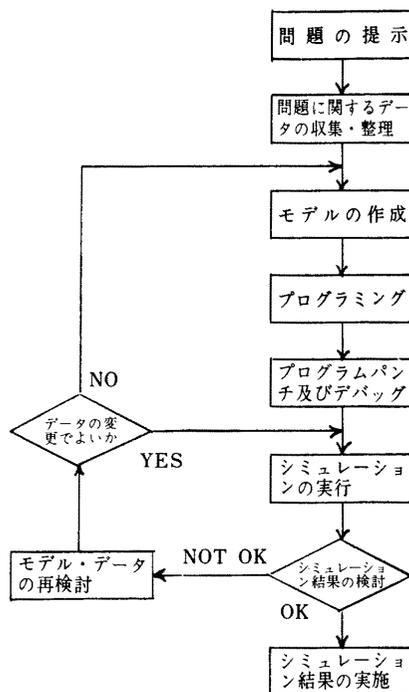


図1 シミュレーションの適用手順

シミュレーションでは数値計算などと異なり、結果が必ずしも唯一に定まらない場合が多いので、試行錯誤的に最適な解へと近づけてゆく。

3. SOLとGPSS/60

SOLとGPSS/60は共に次のような基本的な機能を持っている。

- (1) システム内の変化の単位となるクロックを内蔵している。
- (2) システム内を動きまわるもの（例えば、交通解析における人や車など）として、トランザクション(transaction, trnと略す)がある。
- (3) trnが使用したり、サービスを受けたり、占有したりするものとして設備(equipment)がある。設備には容量が1のfacilityと、容量が2以上のstoreとがある。前者の例として電話、切符売場などがあり、後者の例として駐車場、倉庫などがある。
- (4) モデル全体に有効な変数と各trnに固有な変数とがある。
- (5) trnは優先順位を持っている。同時に多数のtrnが競合するときは、優先順位によって実行順序が決定される。同じ優先順位であれば先着順となる。
- (6) いろいろな乱数を発生させることができる。
- (7) いろいろな統計量を自動的に収集、整理して出力する。

さらに細かい点については、次にあげる例題をSOLとGPSS/60とでプログラミングし、その結果を見ながら説明を進めてゆくことにする。

○ 例題 〈有料駐車場のシミュレーション〉

A. 条件

- ① 営業時間 6～24時
- ② 収容台数 30台
- ③ 料金所 1ヶ所（処理時間 入車時－1分、出車時－2分）
- ④ 入車間隔
6～9時 5～15分
9～16時 3～10分
16～20時 1～5分
20～24時 5～15分
- ⑤ 駐車時間 10～240分
- ⑥ 駐車料金 30分 50円
- ⑦ 満車の時に入ってきた車は待たずに帰る。

B. 調査項目

- ① 入車した車に関するデータ（入車時刻、出車時刻、駐車時間、駐車料金）とその動き
- ② 満車で帰った車の数

- ③ 本日の収入
- ④ 駐車場と料金所の利用状態
- ⑤ 入車時刻、出車時刻、駐車時間のヒストグラム

3. 1 FACOM230-60 SOL

```

SEQ          SOL SOURCE STATEMENT
1          'COMMENT          SOL EXAMPLE          ..
2          'BEGIN
3          'INTEGER          INCOME, NOENTR..
4          'FACILITY          GATE..
5          'STORE          30..GARAGE..
6          'TABLE          (6 'STEP 1 'UNTIL 24) COMEIN, (6 'STEP 1 'UNTIL 24) GOOUT,
7          (6 'STEP 30 'UNTIL 240) PKGTIM..
8          /'PROCEDURE          PRNTNG(ID, N, T, C, P, H)..
9          / 'INTEGER          ID, N, T, C, P, H..
10         / 'BEGIN
11         / 'INTEGER          M..
12         / H=6*T//60.. M=T-(T//60)*60.. CRLF..
13         / 'IF ID 'EOL 1 'THEN
14         / OUTPUT3(200, '30B, *A CAR (* 'ZZD, *) ENTERED AT TIME * 'ZD,
15         / * * 'DD/' 'N, H, M) 'ELSE
16         / OUTPUT5(200, '10B, *A CAR (* 'ZZD, *) LEFT AT TIME * 'ZD, * * ,
17         / DD '5B, *PARKING TIME * 'ZZD, * MIN, * '5B, *CHARGE * ,
18         / ZZD, * YEN* /' 'N, H, M, P, C)..
19         / 'END..
20         'PROCESS          MASTER..
21         'BEGIN
22         INCOME=0.. NOENTR=0..
23         'FJECT.. 'CRLF(4).. 'SPACE(20)..
24         'OUTPUT('** A SIMULATION OF A PAY PARKING LOT **').. 'CRLF(2)..
25         'WAIT 18*60.. 'CRLF(4).. 'SPACE(20)..
26         'OUTPUT('NOT ENTERED CAR', NOENTR).. 'CRLF(4).. 'SPACE(20)..
27         'OUTPUT('TODAY'S INCOME', INCOME, ' YEN')..
28         'STOP..
29         'END..
30         'PROCESS          PARKIN..
31         'BEGIN
32         'INTEGER          CARNB, PT, CHARGE, WT, HR1, HR2..
33         CARNB=0..
34         A R1/L..CARNB=CARNB+1..
35         'IF TIME 'LS 3*60 'THEN WT=RANDI(5,15) 'ELSE
36         'IF TIME 'LS 10*60 'THEN WT=RANDI(3,10) 'ELSE
37         'IF TIME 'LS 14*60 'THEN WT=RANDI(1, 5) 'ELSE WT=RANDI(5,15)..
38         'WAIT WT..
39         'NEW TRN TO GATEIN.. 'GO TO ARRIVL..
40         GATFIN.. 'IF GARAGE 'FULL 'THEN 'BEGIN
41         NOENTR=NOENTR+1.. 'CANCEL.. 'END..
42         'SEIZE GATE.. 'WAIT 1.. 'RELEASE GATE..
43         'ENTER GARAGE,1..
44         / PRNTNG(1, L.LVI(1), L.TIM, 0, 0, L.LVI(5)..
45         'TABULATE HR1 'IN COMEIN..
46         PT=RANDI(10,240)..
47         'WAIT PT..
48         'LEAVE GARAGE,1..
49         'SEIZE GATE.. 'WAIT 2.. 'RELEASE GATE..
50         CHARGE=((PT+30)//30)*50..
51         INCOME=INCOME+CHARGE..
52         / PRNTNG(2, L.LVI(1), L.TIM, L.LVI(3), L.LVI(2), L.LVI(6)..
53         'TABULATE PT 'IN PKGTIM..
54         'TABULATE HR2 'IN GOOUT..
55         'CANCEL..
56         'END..
57         'END..

```

例題のSOLによるプログラムを図2に示す。これを見るとSOLのステートメントはALGOLのそれに極めてよく似ていることがわかる。事実、SOLプログラムはまずSOLトランスレータによりALGOLへ変換され、次にALGOLコンパイラにかけられる。このためSOLプログラムにはALGOLのステートメントを挿入することができる(プログラム中第1コラムが/であるステートメントはALGOLのステートメント)。そこでALGOLで十分に定義されているステートメントについては、ここで詳しく説明することを省略する。また、以下の説明で[n]はプログラム中SEQがnのステートメントを示す。

SOLプログラムはいくつかのプロセスと全体的な宣言とで構成される。プロセスとは一つのまとまった機能を表わすもので、この例ではMASTERとPARKINがある。MASTERはシミュレーションの初期条件を与えたり、シミュレーション時間を決めたりなど全体のコントロールを行なうプロセスであり、PARKINは車が駐車場に来て帰るまでの動きを追うプロセスである。各プロセスではプロセス宣言[20, 30](SEQが20および30のステートメントを参照、以下同様)されるとプロセス自身のtrnが1個発生される。

プログラムの先頭は全体的な宣言の部分で、設備(equipment)と全体変数(global variable)を宣言する。これらはシステム全体で有効であり、変数については参照、書き換えを、設備については占有、解放などをどのプロセス内からでもすることができる。このプログラムでは、まず本日の収入と満車で帰った車の数をカウントするINCOMEとNOENTRを整変数として宣言する[3]。料金所をGATEというfacilityで宣言し[4]、駐車場をGARAGEというstoreで容量30と宣言する[5]。また、統計量を蓄積するために入車時刻、出車時刻、駐車時間をそれぞれCOMEIE、GOOUT、PKGTIMの名前でtable宣言する[6, 7]。COMEIN、GOOUTは6時から24時まで1時間刻み、PKGTIMは0分から240分まで30分刻みでデータを収集する。

プロセスMASTERではtrnはプロセス自身のtrnだけである。このtrnは始めにINCOME、NOENTRを0にし[22]、タイトルをプリントする[23, 24]。次にWAITで18時間待つ[25]。trnがこのwaitステートメントを実行すると、そのtrnのクロックが'WAITの後に指定された算術式の値だけ進められる。このプログラムでは1クロックタイムを1分としているが、これはプログラムを作る人が任意に決めてよい。最後にこのプロセス自身のtrnはNOENTR、INCOMEをプリントし[26, 27]、シミュレーションを終了する[28]。stopステートメントはfacility、store、tableの統計レポートを自動的にプリントし、シミュレーションをストップさせるステートメントである。

次にプロセスPARKINを説明する。このプロセスでは、まず固有変数(local variable)が宣言される[32]。固有変数はそれが宣言されたプロセス内だけに意味を持つもので、そのプロセス内のすべてのtrnに付随する変数である。ここでは入って来た車の通し番号をCARNB、その車の駐車時間

をPT、駐車料金をCHARGE、入車と出車時刻をそれぞれHR 1、HR 2（単位は時）で表わしている[32]。プロセス自身のtrnはまずCARNBを0とし[33]、次にCARNBを1増やす[34]。そしてこのときのクロックに応じて何分か待ち、新しいtrnを発生させる。システムのクロックタイムはTIMEで知ることができる。プロセス自身のtrnはTIMEが3時間以内のときは5～15分、3～10時間のときは3～10分、10～14時間のときは1～5分、それ以上のときは5～15分だけ待つことになる[35～38]。ここでRANDI (n₁, n₂) はn₁からn₂までの間の整乱数を発生させる関数である。プロセス自身のtrnは何分か待った後に、'NEW TRN TO で新しいtrnをラベルGATEINに送り、自分はラベルARRIVLに戻ってこのルーチンを繰り返す[39]。こうしてプロセス自身のtrnにより、CARNBが1の車から、2の車、3の車、…が順次[35～38]で決められた間隔でGATEINに送られることになる。

GATEINに送られた車は、まずGARAGEが一杯かどうかを調べる[40]。'FULLはstoreの全容量使用中を示す設備演算子(equipment operator)で、storeに関してはこの他に'NOT FULL、'EMPTY、'NOT EMPTYがあり、facilityについては'BUSY、'NOT BUSYがある。満車である場合にはNOENTRを1増やし、'CANCELによりこのtrnは消滅される[41]。満車でない場合はGATEを'SEIZEにより占有し、駐車券をもらう時間（1分）待って'RELEASEにより占有を解放する[42]。seizeステートメントを実行したtrnは指定したfacilityが未使用(not busy)であれば、そのfacilityを占有し使用(busy)の状態にして次のステートメントを実行する。facilityがbusyであればそのtrnはnot busyになるまで待ちに入る。GATEを通過したtrnは'ENTERでGARAGEに入る[43]。enterステートメントを実行したtrnはstoreの名前の後に書いてある算術式の値だけstoreを使用するが、もしそのstoreにそれだけの空きがない場合は空きができるまでそのtrnは待ちに入る。GARAGEに入ったtrnは入車時刻HR 1をtable COMEINに記入する[45](HR 1の計算は後述)。次に駐車時間PTを10～240分の整乱数で与え[46]、その時間だけ待って(駐車して)[47]、'LEAVEによりGARAGEを出る[48]。leaveステートメントはenterステートメントの逆で、storeの名前の後の算術式の値だけstoreの使用を解放する。駐車を終ったtrnはGATEを占有し、駐車料金を払う時間（2分）待ってGATEを解放する[49]。その後駐車料金を計算し[50]、それを本日の収入に加える[51]。最後に、駐車時間PTと出車時刻HR 2をそれぞれtable PKGTIM、GOOUTに記入して[53, 54]、このtrnは消滅する[55]。

次に、途中に挿入されているALGOLステートメント[8～19, 44, 52]を説明する。これはALGOLのprocedure[8～19]とその呼び出し[44, 52]で、入車時と出車時に必要なプリントを行なうものである。仮パラメータIDが1のとき入車時を、2のとき出車時を示し、NはCARNB、Tはシステムクロック、CはCHARGE、PはPT、HはHR 1またはHR 2に対応している[9]。[12]でTよりHに時、

Mに分を入れ、IDの値によって[14]あるいは[16]のOUTPUT文で出力する。このprocedureを呼び出すときは、実パラメータを注意して与えなければならない。すなわちSOLプログラムがSOLトランスレータによりALGOLに変換されると、各固有変数は配列L.LVIの要素で表わされる。このためCARNB、PT、CHARGE、…はALGOLの段階でこの順にL.LVI(1)、L.LVI(2)、L.LVI(3)、…となっており、ALGOLステートメントでの実パラメータとするにはこの名前を用いなければならない[44, 52]。また、システムクロックを示すTIMEもALGOLではEXTERNALなINTEGER PROCEDURE L.TIMとなっている。このようにALGOLステートメントをSOLプログラム中に挿入するときは、SOLとALGOLの間の名前の違いに充分に気をつける必要がある。またSOLプログラムにALGOLステートメントを挿入する場合、プログラムは26型で穿孔されていなければならない。

ここでは出力をスマートにするためにALGOLのprocedureを用いたが、[24, 26, 27]のようにSOLに用意されている標準のoutputステートメントを用いれば、例えば[44]のALGOLステートメントは

```
'OUTPUT("A CAR (" , CARNB, ") ENTERED AT TIME", TIME),,
```

となる。同様に入力についても標準のinputステートメントがある。

これまで説明をしてきたSOLプログラムを実行した結果を図3に示す。まず図3-aを見ると、1番の車は6時6分、2番の車は6時12分、…と入車して、2番の車は98分駐車のもの7時52分に駐車料金200円を払って出車したことがわかる。これは車の動きを知るために、ALGOLのprocedureで入車時と出車時にプリントさせた分である。また、最後には満車で帰った車が18台、本日の収入が32,850円とプロセスMASTERのoutputステートメントで出力されている。次に、図3-bはプロセスMASTERのstopステートメントで自動的にプリントされた統計レポートである。これによると facility GATEは延べ454分占有され、利用率は約42%であり、store GARAGEには最高30台、平均17.3台駐車されていたことがわかる。table COMEIN、GOOUTについては、UPPER LIMITの欄で6~24時を示し、NUMBERにその時刻に入車あるいは出車した車の台数が示されている。table PKG-TIMではUPPER LIMITの欄で30分刻みの駐車時間を示し、NUMBERにその時間駐車した車の台数が示されている。これらのtableによると、入車間隔と駐車時間は条件を満たしていることがわかる。

** A SIMULATION OF A PAY PARKING LOT **					
A CAR (1)	ENTERED AT TIME	6.06			
A CAR (2)	ENTERED AT TIME	6.12			
A CAR (3)	ENTERED AT TIME	6.25			
A CAR (4)	ENTERED AT TIME	6.35			
A CAR (5)	ENTERED AT TIME	6.46			
A CAR (6)	ENTERED AT TIME	6.56			
A CAR (7)	ENTERED AT TIME	7.04			
A CAR (8)	ENTERED AT TIME	7.12			
A CAR (9)	ENTERED AT TIME	7.25			
A CAR (10)	ENTERED AT TIME	7.37			
A CAR (11)	ENTERED AT TIME	7.47			
A CAR (2)	LEFT AT TIME	7.52	PARKING TIME	98 MIN.	CHARGE 200 YFN
A CAR (7)	LEFT AT TIME	7.58	PARKING TIME	52 MIN.	CHARGE 100 YFN
A CAR (12)	ENTERED AT TIME	8.00			
A CAR (13)	ENTERED AT TIME	8.08			
A CAR (5)	LEFT AT TIME	8.15	PARKING TIME	87 MIN.	CHARGE 150 YFN
途中省略					
A CAR (150)	LEFT AT TIME	23.41	PARKING TIME	233 MIN.	CHARGE 400 YFN
A CAR (177)	ENTERED AT TIME	23.46			
A CAR (156)	LEFT AT TIME	23.56	PARKING TIME	219 MIN.	CHARGE 400 YFN
A CAR (178)	ENTERED AT TIME	23.59			
NOT ENTERED CAR 18					
TODAY'S INCOME 32850 YEN					

図3-a

F A C O M 230-60 S O L R E P O R T

12-23.45

PAGE 1

STIMULATION TIME 1080
 FACILITY NAME CUMU.OCCUPANCY AVERAGE CURRENT STATUS
 GATE 454 0.4204 NOT BUSY

STORE NAME CAPACITY CURRENT USE MAXIMUM USED CUMU.USED MEAN OF USED
 GARAGE 30 13 30 18670 17.2870

TABLE NAME SUM OF ENTRY AVERAGE MAXIMUM MINIMUM VARIANCE STD.DEVIATION
 COMEIN 2367.0000 14.7938 23.0000 6.0000 20.9387 4.5759

UPPER LIMIT	NUMBER	PERCENT	CUMU.PERCENT
6.0000	6	3.75	3.75
7.0000	5	3.13	6.88
8.0000	6	3.75	10.63
9.0000	10	6.25	16.88
10.0000	8	5.00	21.88
11.0000	9	5.63	27.50
12.0000	9	5.63	33.13
13.0000	8	5.00	38.13
14.0000	9	5.63	43.75
15.0000	8	5.00	48.75
16.0000	16	10.00	58.75
17.0000	18	11.25	70.00
18.0000	15	9.38	79.37
19.0000	8	5.00	84.38
20.0000	6	3.75	88.13
21.0000	7	4.38	92.50
22.0000	6	3.75	96.25
23.0000	6	3.75	100.00
24.0000	0	0.00	100.00
OVER	0	0.00	100.00
TOTAL	160		100.00

TABLE NAME SUM OF ENTRY AVERAGE MAXIMUM MINIMUM VARIANCE STD.DEVIATION
 GOOUT 2379.0000 16.1837 23.0000 7.0000 19.2520 4.3877

UPPER LIMIT	NUMBER	PERCENT	CUMU.PERCENT
6.0000	0	0.00	0.00
7.0000	2	1.36	1.36
8.0000	5	3.40	4.76
9.0000	8	5.44	10.20
10.0000	7	4.76	14.97
11.0000	5	3.40	18.37
12.0000	9	6.12	24.49
13.0000	8	5.44	29.93
14.0000	4	2.72	32.65
15.0000	10	6.80	39.46
16.0000	10	6.80	46.26
17.0000	15	10.20	56.46
18.0000	14	9.52	65.99
19.0000	10	6.80	72.79
20.0000	11	7.48	80.27
21.0000	13	8.84	89.12
22.0000	8	5.44	94.56
23.0000	8	5.44	100.00
24.0000	0	0.00	100.00
OVER	0	0.00	100.00
TOTAL	147		100.00

TABLE NAME SUM OF ENTRY AVERAGE MAXIMUM MINIMUM VARIANCE STD.DEVIATION
 PKGTIM 17668.0000 120.1905 239.0000 10.0000 4300.4401 65.5777

UPPER LIMIT	NUMBER	PERCENT	CUMU.PERCENT
0.0000	0	0.00	0.00
30.0000	16	10.88	10.88
60.0000	20	13.61	24.49
90.0000	17	11.56	36.05
120.0000	22	14.97	51.02
150.0000	22	14.97	65.99
180.0000	17	11.56	77.55
210.0000	17	11.56	89.12
240.0000	16	10.88	100.00
OVER	0	0.00	100.00
TOTAL	147		100.00

☒ 3-b

3. 2 FACOM230-60 GPSS/60

```

JOB# 1
GPSS SOURCE LIST
-----
BLOCK NO  LOC  NAME  A.B.C.D.E.F.G  COMMENT
*
* ***** A SIMULATION OF A PAY PARKING LOT *****
*
* EQUIPMENTS
* 1 STORAGE 30 GARAGE ( CAPACITY 30 )
* FACILITY 1 GATE
*
* DATA
* SAVEVALUE 1 NUMBER OF NOT ENTERED CARS
* SAVEVALUE 2 TODAY'S INCOME
* PARAMETER 1 ARRIVING TIME
* PARAMETER 2 PARKING TIME
* PARAMETER 3 CHARGE
* PARAMETER 4 DEPARTING TIME
*
* TABLES
* 1 TABLE P1.6.1.18 TABLE FOR ARRIVING TIME
* 2 TABLE P4.6.1.18 TABLE FOR DEPARTING TIME
* 3 TABLE P7.0.30.8 TABLE FOR PARKING TIME
*
* FUNCTION FOR ARRIVING INTERVAL
* 1 FUNCTION RN(1)*C2
* 0.0 5 1.0 16
* 2 FUNCTION RN(1)*C2
* 0.0 3 1.0 11
* 3 FUNCTION RN(1)*C2
* 0.0 1 1.0 8
* 4 FUNCTION C1.E4
* 18.0 FN1 600 FN2 840 FN3 1080 FN1
*
* FUNCTION FOR PARKING TIME
* 5 FUNCTION RN(1)*C2
* 0.0 10 1.0 240
*
* VARIABLE FOR CLOCK
* 1 VARIABLE C1/60*6
*
* VARIABLE FOR CHARGE
* 2 VARIABLE P2/30*50+50
*
* 1 GENERATE 1.FN4 GENERATE ARRIVING CARS
* 2 TRACE TRACE TRNS
* 3 GATE SF 1.IN TEST IF GARAGE IS FULL
* 4 SAVEVALUE 1+1 ADD 1 TO NUMBER OF NOT ENTERED CARS
* 5 TERMINATE
*
* 6 IN SFIZE 1 SEIZE GATE
* 7 ADVANCE 1 TIME FOR RECEIVING A TICKET
* 8 RLEASEF 1 RELEASE GATE
* 9 ENTER 1 ENTER GARAGE
* 10 ASSIGN 1.V1 ASSIGN ARRIVING TIME TO P1
* 11 TABULATE 1 TABULATE ARRIVING TIME
* 12 ASSIGN 2.FN5 ASSIGN PARKING TIME
* 13 ADVANCE P2 TIME FOR PARKING
* 14 TABULATE 3 TABULATE PARKING TIME
* 15 LEAVE 1 LEAVE GARAGE
* 16 ASSIGN 3.V2 ASSIGN CHARGE TO P3
* 17 SAVEVALUE 2+P3 ADD CHARGE TO TODAY'S INCOME
* 18 ASSIGN 4.V1 ASSIGN DEPARTING TIME TO P4
* 19 TABULATE 2 TABULATE DEPARTING TIME
* 20 SFIZE 1 SEIZE GATE
* 21 ADVANCE 2 TIME FOR PAYING CHARGE
* 22 RELEASE 1 RELEASE GATE
* 23 TERMINATE
*
* 24 GENERATE 1080 GENERATE CONTROL TRN AT 24:00
* 25 TERMINATE 1 SIMULATION END
*
* / START 1...CHAIN
* / END
    
```

例題のGPSS/60によるプログラムとそのブロックダイアグラムを図4、図5に示す。GPSS/60ではシステムの構造をブロックダイアグラムで表現する。各々のブロックはその書き方が定められており、1個のブロックはプログラムの1つのステートメントに対応している。プログラムのカード形式は、第1～6コラムがlocation、第7コラムがcontinuation、第8～18コラムがoperation、第19～72コラムがoperandとなっており、operand欄は空白が出現すればそれ以下はcommentとみなされる。また、第1コラムが*のカードはcommentカード、/のカードはcontrolカードである。

まずプログラムの先頭でstorage、table、function、variableの定義を行なう。facility、savevalue、parameterは定義しなくてもあらかじめ必要な領域が確保されているが、ここではcommentカードでそれらの表わす意味を示している。設備としては、駐車場をstorage1で大きさ30と定義し、facility1は料金所として用いる。次にSOLでの全体変数と固有変数に対応するものとして、それぞれsavevalueとparameterがあり、savevalue 1, 2で満車で帰った車の数と本日の収入をカウントする。parameter

1, 2, 3, 4はそれぞれ入車時刻、駐車時間、駐車料金、出車時刻を示すものである(GPSS/60ステートメントではparameter i は P_i と表わす)。table1は入車時刻(P_1)を集計するtableで6時から1時間刻みで18時間のヒストグラムを作る。table2は出車時刻(P_4)、table3は駐車時間(P_2)のためのtableである。これらの設備、変数、tableはいずれもSOLのそれに対応していたが、GPSS/60ではSOLのようにプログラム中に任意の数式を書くことができないため、functionとvariableにより種々の計算を行なう。このプログラムではfunction 1, 2, 3, 4, 5とvariable 1, 2を定義している(parameterと同様にfunction i は FN_i 、variable i は V_i と表わす)。 FN_1 , FN_2 , FN_3 はそれぞれ5～15, 3～10, 1～5の整乱数を発生するfunctionで、 FN_4 はこれらを組み合わせて18時間(1080分)の入車間隔を作るfunctionである。また FN_5 は10～240分の駐車時間のためのfunctionである。 V_1 は

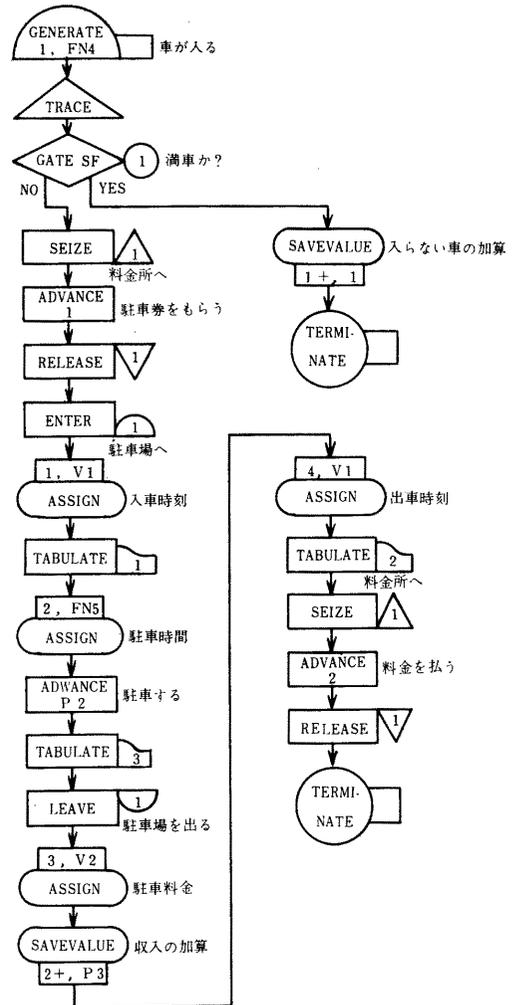


図5

システムのクロック(C1)よりその時の時刻(単位は時)を計算するもので、V2は駐車時間(P2)より駐車料金を計算するvariableである。variableではカッコの使用は認められない。

種々の定義が終わったのち、実際のプログラムに入る。先に述べたように GPSS/60では定義以外の一般のステートメントはブロックダイアグラムの各ブロックと一対一に対応している。このためステートメントはブロックと呼ばれ、プログラム内の各ブロックは先頭から順に番号(BLOCK NO)がつけられている。以下の説明で[n]はこのBLOCK NOがnのブロックを示すことにする。このプログラムのtrnはSOLのプロセスPARKINと同様に車を表わす。まず、GENERATEブロックで入ってくる車をFN4で示す間隔で発生させる[1]。operand欄の1, FN4は $1 \times FN4$ の間隔を示すが、第2番目がfunctionでない場合、例えば「GENERATE 5, 3」などのときは 5 ± 3 の間隔で発生させることを意味する。次にTRACEブロックで各trnの流れてゆくブロックを追跡する[2]。GPSS/60では任意の文を出力させることが非常に困難なので、本来デバッグ用に作られたこのTRACEブロックで各trn(車)の動きを知ることにする。入ってきた車はGATEブロックで駐車場が満車か否かを調べる[3]。SFはstorageがfullのとき真になる論理変数、1はstorage番号、INはSFが真でないときに進むブロックのlocationである。SFが真であるときは次のブロックに進む。満車のときは次のSAVEVALUEブロックで帰った車をカウントし[4]、TERMINATEブロックでその車は消滅される[5]。SAVEVALUEブロックの1+, 1はsavevalue 1に1を加えることを意味する。一方、駐車場に空きがあった場合はlocation INで示されるSEIZEブロックに入り料金所(facility 1)を占有する[6]。facilityがbusyのときはSOLと同様にnot busyになるまでそのtrnは待ちに入る。次にSOLのwaitステートメントに対応するADVANCEブロックで駐車券をもらう時間(1分)待ち[7]、RELEASEブロックで料金所を解放する[8]。料金所を通過した車はENTERブロックで駐車場(storage 1)に入る[9]。storageがfullのときもSOLと同様にnot fullになるまでそのtrnは待ちに入る。入車と同時にASSIGNブロックでV1で計算される入車時刻をparameter 1にセットし[10]、それをTABULATEブロックでtable 1に記入する[11]。続いて、同じくASSIGNブロックでFN5で計算される駐車時間(10~240分)をparameter 2にセットし[12]、ADVANCEブロックでその時間(P2)だけ待つ(駐車する)[13]。駐車を終った車はTABULATEブロックで駐車時間(P2)をtable 3に記入し[14]、LEAVEブロックで駐車場(storage 1)を出る[15]。そののち、ASSIGNブロックでV2で計算される駐車料金をparameter 3にセットし[16]、SAVEVALUEブロックでそれを本日の収入(savevalue 2)に加算する[17]。また、ASSIGNブロックでV1で計算される出車時刻をparameter 4にセットし[18]、それをTABULATEブロックでtable 2に記入する[19]。続いてSEIZEブロックで料金所(facility 1)を占有し[20]、ADVANCEブロックで駐車料金を払う時間(2分)待つ[21]、RELEASEブロックで料金所を解放する[22]。これで車は駐車場を出たことに

なり、TERMINATEブロックで消滅される[23]。最後にSOLのプロセスMASTERに相当するものとして、1080分(18時間)後に1つのtrnを発生させ[24]、それをすぐに消滅させる[25]。このTERMINATEブロックのoperand欄には1が記入されているが、trnがこのブロックを通過する度に次のSTARTカードのoperand欄に記入されている数字(termination count)からこの数が引かれ termination countが0以下になったときシミュレーションがstopされる。この場合は共に1であるので、一度[25]のTERMINATEブロックをtrnが通過するとシミュレーションがstopされる。また、例えば[23]のTERMINATEブロックに1を指定し、STARTカードのtermination countを500としておけば、駐車した車が500台でstopすることもできる。

***** SIMULATION START *****				JOB=(1)	-(GPSS/60)-	
TRN NO.=	1	BLOCK (FROM 2 TO 3)	CLOCK=	6	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 3 TO 6)	CLOCK=	6	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 6 TO 7)	CLOCK=	6	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 7 TO 8)	CLOCK=	7	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 8 TO 9)	CLOCK=	7	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 9 TO 10)	CLOCK=	7	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 10 TO 11)	CLOCK=	7	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 11 TO 12)	CLOCK=	7	TERMINATION COUNT=	1
TRN NO.=	1	BLOCK (FROM 12 TO 13)	CLOCK=	7	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 2 TO 3)	CLOCK=	12	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 3 TO 6)	CLOCK=	12	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 6 TO 7)	CLOCK=	12	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 7 TO 8)	CLOCK=	13	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 8 TO 9)	CLOCK=	13	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 9 TO 10)	CLOCK=	13	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 10 TO 11)	CLOCK=	13	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 11 TO 12)	CLOCK=	13	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 12 TO 13)	CLOCK=	13	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 2 TO 3)	CLOCK=	27	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 3 TO 6)	CLOCK=	27	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 6 TO 7)	CLOCK=	27	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 7 TO 8)	CLOCK=	28	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 8 TO 9)	CLOCK=	28	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 9 TO 10)	CLOCK=	28	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 10 TO 11)	CLOCK=	28	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 11 TO 12)	CLOCK=	28	TERMINATION COUNT=	1
TRN NO.=	4	BLOCK (FROM 12 TO 13)	CLOCK=	28	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 13 TO 14)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 14 TO 15)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 15 TO 16)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 16 TO 17)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 17 TO 18)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 18 TO 19)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 19 TO 20)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 20 TO 21)	CLOCK=	39	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 21 TO 22)	CLOCK=	41	TERMINATION COUNT=	1
TRN NO.=	3	BLOCK (FROM 22 TO 23)	CLOCK=	41	TERMINATION COUNT=	1

P28

図6-a

これまで説明してきたGPSS/60のプログラムを実行した結果を図6に示す。まず図6-aはTRACEブロックで各trnの追跡を行なった結果である。これを見ると、例えば1番のtrn(車)はシステムのクロックが6のときにブロック2から3へと移動したことがわかる。次に図6-bにstorage、facility、table、savevalueに関する統計レポートを示す。facility 1(料金所)は利用率約43%であり、storage 1(駐車場)は最高30台、平均18.6台駐車されていたことがわかる。また、table 1(入車時刻)とtable 2(出車時刻)はUPPER LIMITの欄で6~24時を示し、OBSERVED FREQUENCYの欄にその時刻に入車あるいは出車した車の台数が示されている。table 3(駐車時間)はUPPER LIMITの欄で0~240分を示し、OBSERVED FREQUENCYの欄にその時間駐車した車の台数が示されている。これらのtableによると、入車間隔と駐車時間は条件を満たしていることがわかる。savevalueについては、savevalue 1に満車で帰った車の台数は24台、savevalue 2に本日の収入は35650円と示され

```

JOB.NO= 1 ***** STATISTICAL PRINT OUT FOR FACILITIES *****
FACILITY NO. 1 AVERAGE UTILIZATION 0.4315 NUMBER ENTRIES 315 AVFRAGF TIME/TRANS 1.48 SEIZING TRANS.NO. 0 PREEMPTING TRANS.NO. 0

JOB.NO= 1 ***** STATISTICAL PRINT OUT FOR STORAGE *****
STORAGE NUMBER 1 CAPACITY 30 AVERAGE CONTENTS 1A.55 AVERAGE UTILIZATION 0.6185 ENTRIES 164 AVERAGE TIME/TRANS. 122.18 CURRENT CONTENTS 13 MAXIMUM CONTENTS 30

JOB.NO= 1 ***** STATISTICAL PRINT OUT FOR TABLES *****
TABLE ENTRIES 164 MEAN ARGUMENTS 14.720 STANDARD DEVIATION 4.541 SUM OF ARGUMENTS 2414.000 OVERFLOW FREQUENCY 0 AVERAGE VALUE OF OVERFLOW 0.0

TABLE.NO. UPPER LIMIT OBSERVED FREQUENCY PERCENT OF TOTAL CUMULATIVE PERCENTAGE CUMULATIVE REMAINDER MULTIPLE OF MEAN DEVIATION FROM MEAN
1 6 5 3.05 3.0 97.0 0.0 -3.242
1 7 6 3.66 6.7 93.3 0.476 -1.700
1 8 6 3.66 10.4 89.6 0.543 -1.480
1 9 9 5.49 15.9 84.1 0.611 -1.260
1 10 10 6.10 22.0 78.0 0.679 -1.039
1 11 9 5.49 27.4 72.6 0.747 -0.819
1 12 11 6.71 34.1 65.9 0.815 -0.599
1 13 11 6.71 40.9 59.1 0.883 -0.379
1 14 8 4.88 45.7 54.3 0.951 -0.158
1 15 9 5.49 51.2 48.8 1.019 0.062
1 16 17 10.37 61.6 38.4 1.087 0.287
1 17 12 7.32 68.9 31.1 1.155 0.507
1 18 13 7.93 76.8 23.2 1.223 0.727
1 19 12 7.32 84.1 15.9 1.291 0.943
1 20 9 5.49 89.6 10.4 1.359 1.163
1 21 5 3.05 92.7 7.3 1.427 1.383
1 22 7 4.27 97.0 3.0 1.495 1.603
1 23 5 3.05 100.0 0.0 1.563 1.824
REMAINING FREQUENCIES ARE ALL ZERO

JOB.NO= 1 ***** STATISTICAL PRINT OUT FOR TABLES *****
TABLE ENTRIES 151 MEAN ARGUMENTS 16.132 STANDARD DEVIATION 4.173 SUM OF ARGUMENTS 2436.000 OVERFLOW FREQUENCY 0 AVERAGE VALUE OF OVERFLOW 0.0

TABLE.NO. UPPER LIMIT OBSERVED FREQUENCY PERCENT OF TOTAL CUMULATIVE PERCENTAGE CUMULATIVE REMAINDER MULTIPLE OF MEAN DEVIATION FROM MEAN
2 6 1 0.66 0.7 99.3 0.0 -3.866
2 7 1 0.66 1.3 98.7 0.434 -2.189
2 8 3 1.99 3.3 96.7 0.434 -1.949
2 9 3 1.99 9.3 90.7 0.558 -1.709
2 10 3 1.99 11.3 88.7 0.620 -1.470
2 11 9 5.96 17.2 82.8 0.682 -1.230
2 12 6 3.97 21.2 78.8 0.744 -0.990
2 13 10 6.62 27.8 72.2 0.806 -0.751
2 14 12 7.95 35.8 64.2 0.868 -0.511
2 15 10 6.62 42.4 57.6 0.930 -0.271
2 16 10 6.62 49.0 51.0 0.992 -0.032
2 17 11 7.28 56.3 43.7 1.054 0.208
2 18 12 7.95 64.2 35.8 1.116 0.448
2 19 17 11.26 75.5 24.5 1.178 0.687
2 20 15 9.93 85.4 14.6 1.240 0.927
2 21 5 3.31 88.7 11.3 1.302 1.166
2 22 13 8.61 97.4 2.6 1.364 1.406
2 23 4 2.65 100.0 0.0 1.426 1.646
REMAINING FREQUENCIES ARE ALL ZERO

JOB.NO= ***** STATISTICAL PRINT OUT FOR TABLES *****
TABLE ENTRIES 151 MEAN ARGUMENTS 126.424 STANDARD DEVIATION 68.565 SUM OF ARGUMENTS 19090.000 OVERFLOW FREQUENCY 21 AVERAGE VALUE OF OVERFLOW 224.8095

TABLE.NO. UPPER LIMIT OBSERVED FREQUENCY PERCENT OF TOTAL CUMULATIVE PERCENTAGE CUMULATIVE REMAINDER MULTIPLE OF MEAN DEVIATION FROM MEAN
3 0 0 0.0 0.0 100.0 0.0 -1.844
3 30 20 13.25 13.2 86.8 0.237 -1.406
3 60 12 7.95 21.2 78.8 0.475 -0.969
3 90 13 8.61 29.8 70.2 0.712 -0.531
3 120 28 18.54 48.3 51.7 0.949 -0.094
3 150 17 11.26 59.6 40.4 1.186 0.344
3 180 17 11.26 70.9 29.1 1.424 0.781
3 210 23 15.23 86.1 13.9 1.661 1.219

JOB.NO= 1 ***** STATISTICAL PRINT OUT FOR SAVE VALUE *****
NO=SAVEVALUE NO.
NO VALUE NO VALUE NO VALUE NO VALUE NO VALUE
1 24 2 35650 3 0 4 0 5 0
6 0 7 0 8 0 9 0 10 0
11 0 12 0 13 0 14 0 15 0
    
```

図 6-b

ている。

4. おわりに

簡単な例題を示してSOLとGPSS/60のプログラムを紹介したが、表面的にプログラムを追って行くだけに終わってしまった。紹介したステートメント以外にもSOLとGPSS/60はいろいろな機能をもっているし、また実際にプログラムを作製する場合は、trnをコントロールするルーチンやクロックの進め方などシステムの中核となる部分についても知っておく必要がある。これらの点については紙面の都合上その説明を割愛したが、プログラム作製にあたっては十分にマニュアルを読まれることを希望する。

参考文献

- (1) FACOM 230-60 SOL解説書 (230/60-204-001-1)
- (2) FACOM 230-60 GPSS/60 解説書 (I) (230/60-204-002-1)