

シミュレーション用語について* (II)

関根智明** 反町洋一*** 園部昭夫***

〔付録 II〕 SIMSCRIPT

SIMSCRIPT (Simulation Scriptor)

あらし

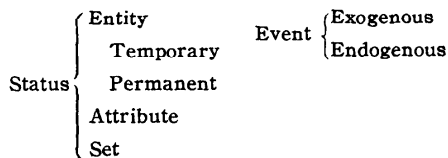
SIMSCRIPT は RAND Corp. (USA) で開発されたシミュレーション用のプログラミング言語であり、現在 USA では多くの分野でその利用についての試みがなされているようである。

SIMSCRIPT ではシミュレーションの対象となるシステムの状態を Entity, Attribute, Set で記述する。

ここで Entity とはシステムの構成要素を、Attribute は構成要素の状態を表わす変数を表わし Entity の Set とは文字どおり Entity の集りである。これらの Entity, Attribute, Set で表わされたシステムの挙動について、すなわち時間的にどのようにシステムの状態が修正されてゆくかについては Event Routine が用いられる。

Event Routine では、そこでシステムの構成要素の状態がどのように修正されるかについて記述すれば良いわけで、異なる Event はいくつあっても良くシミュレーション中に SIMSCRIPT は自動的に Timing Routine からシミュレーションの時間についての情報を得て、適当な道順に Event Routine を呼んでゆく。

この Event Routine の中にもシステムの内部の状況によって引き起される Endogenous Event とシステムの外部から時間を指定して引き起こす Exogenous Event があり、これらの概念は



のようにまとめられ Temporary Entity, Permanent

* On Simulation Language (II), by Tomoharu Sekine (Keio University), Yōichi Sorimachi and Akio Sonobe (Mitsubishi Atomic Power Ind., Inc.)

** 慶応義塾大学 *** 三菱原子力工業 (株)

Entity のあつかい方について例により説明する。

シミュレーションプログラムの例

工場は 10 個の machine group からなる。たとえば machine group 1 はせん盤、machine group 2 はボール盤などで各 group 内での機械はすべて性能は同一である。注文はシミュレーション中の任意の時点に外から工場に到着する。各仕事には処理を受ける machine group の技術的順序と各 machine group での処理時間が指定されている。ある machine group で処理された注文は、指定されている次の machine group へ行き、もし空いている機械があればそこで処理を受け、空いている機械がなければその machine group の待ち行列の中に入る。待ち行列にたまっている仕事を処理して行くための優先規則は先着順とする。

シミュレーションで求める結果は、注文が工場内にある平均時間、各 machine group での行列の長さである。

Status を記述するためにこの例で使われる Entity は

Permanent Entity

MG ; machine group

Temporary Entity

ORDER; 注文

DESTN; 処理順序

EPROC; 処理の終りを表わす。

Endogenous Event のための

Event Notice

Entity の集まりである Set としては、

QUE(MG); 各 machine group での注文の待ち行列の set (Entity ORDER の集合)。この Set を有する Entity は MG である。

ROUT(ORDER); 各注文の処理順序を表わす Set. Entity DESTN の集合。

各 Entity はその Attribute の値によって数量的に記述されるわけだが

NOAVL(MG); MG で使用可能な機械台数

DATE(ORDER); 注文の到着日時

PTIME(DESTN); 指定された MG での処理時間

NINQ(MG); MG の待ち行列の長さ

MGDST(DESTN); 処理される MG 名

これらの Entity, Attribute, Set, は SIMSCRIPT の Definition form によって定義される (第 1 図).

第 1 図をみると各 Entity の Attribute の中に上で説明した以外に

FROUT, LROUT, SROUT

SQUE, FQUE, LQUE

なる Attribute が現われるが, これは Set を構成し, 各種の Set operation のために必要な Attribute である.

このようにして記述された system の status は二つの Event によって修正されシミュレートされる.

Exogenous Event ORDRIN

; 工場の外部から注文の到着という Event を表わす.

Endogenous Event ERROC

; 工場の内部で起る Event, ある machine group の注文処理が終わった Event を表わす.

そのほかに二つの Exogenous Event

Exogenous Event ANALYZ

; 結果を分析する Event

Exogenous Event ENDSIM

; シミュレーションを終了するための Event

これらの Event は SIMSCRIPT program の Event List に登録され (第 2 図), SIMSCRIPT により自動的に作られる Timing Routine によって Simulated Time が記録され, ある Event routine の実行が終わると次の最も近い Event の起きる時刻まで自動的に時刻が進められ, Event を起し, Status が修正され, シミュレーションが進められていくのである.

STATEMENT NUMBER	Continuation	STATEMENT
1 2	5 6 7	77
EVENTS		
3 EXOGENOUS		
ORDFIN (1)		
ANALYZ (2)		
ENDSIN (3)		
1 ENDOGENOUS		
EPROC		
END		

第 2 図 SIMSCRIPT Event List の例

SIMSCRIPT Program の説明

前節で説明した job shop simulation の例で Program の一部の説明を行なう.

EXOGENOUS EVENT ORDRIN

; ORDRIN と呼ばれる Exogenous Event を述べる Event Routine を示す.

TEMPORARY SYSTEM VARIABLES													PERMANENT SYSTEM VARIABLES													SETS				
TEMPORARY AND EVENT NOTICE ENTITIES													ATTRIBUTES																	
NAME	RECORD SIZE												NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER	NAME	ARRAY NUMBER
	1	2	3	4	5	6	7	8	9	10	11	12																		
T ORDERA													T DATE	1	/	F	1	CUMCT	0	/	F					QUE I				
													T FROUT	2	/	I	2	NRDR	0	/	F					ROUT I				
													T LROUT	3	/	I	3	GRAND	0	/	F									
													T SQUE	4	/	I	4	LAST	0	/	F									
E EPROC													N MGPROC	3	/	I	5	MCT	0	/	F									
													N MGRPT	4	/	I	6	MG	E	/	I									
T DESTN													T MGDST	1	/	F	7	DAVL	1	/	I									
													T PTIME	2	/	F	8	FQUE	1	/	I									
													T SROUT	3	/	I	9	LQUE	1	/	I									
																	10	INQ	1	/	F									
																	11	CUMQ	1	/	F									
																	12	TNQ	1	/	F									
																	13	MEANQ	1	/	F									

第 1 図 SIMSCRIPT の Definition form

```

CREATE ORDER
; Entity ORDER の Attribute を記憶するための
Record を作る. ORDER にこの Entity の
Identification Number が与えられる.
LET DATE(ORDER)=TIME
; ORDER の Attribute DATE を注文の到着の
日付の値にする.
READ N
FORMAT (I 4)
DO TO 10, FOR I=(1)(N)
; N に注文の工程数が読み込まれる.
READ MGDST(DESTN), PTIME(DESTN)
FORMAT(I 4, H 3.2)
; Entity DESTN の二つの attribute の値を読
み込む.
FILE DESTN IN ROUT(ORDER)
; Entity DESTN を set ROUT の中に file す
る.
10 LOOP
CALL ARRVL(ORDER)
; 注文の機械グループへの到着を記述する. Sub-
routine ARRVL を call する.
RETURN
; control を Timing Routine にもどす.
END

ENDOGENOUS EVENT EPROC
; Endogenous Event EPROC の routine を示
す.

STORE ORDRP(EPROC) IN ORDER
; EPROC の attribute ORDRP(EPROC),
MGPRC(EPROC) の値を
STORE MGPRC(EPROC) IN MG
; ORDER, MG, に入れる.
DESTROY EPROC
; EPROC の Record を reset し, この部分が
以下に使用可能とする.
C —DISPOSITION OF THE ORDER—
; コメント
IF ROUT(ORDER) IS EMPTY, GO TO 10
; 処理工程が残っていない時は statement 10 に
行く.
CALL ARRVL(ORDER)
; 未処理工程がある場合は subroutine ARRVL

```

```

を呼ぶ.
GO TO 20
10 LET CUMCT=CUMCT+TIME-DATE
(ORDER)
; 全処理時間の計算
LET NORDR=NORDR+1.0
; 処理した注文数
DESTROY ORDER
; 工程処理の終了した注文について Entity OR-
DER の record を reset する.
C —DISPOSITION OF THE MACHINE—
; コメント
20 IF QUE(MG) IS EMPTY, GO TO 30
; 機械グループに待ち行列がない時は statement
30 へゆく.
REMOVE FIRST ORDER FROM QUE(MG)
; 待ち行列の先頭にある Order を待ち行列から
取り除く.
CALL ALLOC(MG, ORDER)
; subroutine ALLOC を call する.
ACCUMULATE NINQ(MG) INTO CUMQ(MG)
-SINCE TMQ(MG), POST NINQ(MG)-1.0
; 累積待ち行列の計算
RETURN
30 LET NOAVL(MG)=NOAVL(MG)+1
; 待ち行列がない時は使用可能な機械台数を 1 台
増やす.
RETURN
END

```

Subroutine ARRVL, ALLOC, 統計をとるための Exogenous Event ANALYZ, Simulation の終了のための Exogenous Event ENDSIM Report Generator については説明を省略した.

〔付録 III〕 DYNAMO

DYNAMO (DINAmic MOdel)

近年 Jay. W. Forrester (MIT) の著 “Industrial Dynamics” がシミュレーション分野で注目されているが, この種の計算機用プログラムシステムとしては 1958 年に SIMPLE (IBM 704), 次いで 1959 年にこの DYNAMO (Dr. Phillis Fox & Alexander L. Pugh III により開発された) が登場した.

その名のように組織体のダイナミックな動きをとら

える型のシミュレーションに有効で、たとえば経済予測、会社の膨脹あるいは在庫管理の問題など、多くの問題に威力を発揮すると思われる。

DYNAMO の特長を簡易に述べると

- 1) compiler でなく generator である。
- 2) したがって方程式の順序が 2, 3 の例を除いて任意である。
- 3) 一般の計算機用言語 (FORTRAN, ALGOL) に不慣れた人にも理解し易い。
- 4) output を表とグラフでリストさせる手続が容易である。

反面、実際にモデルを組む立場から

- 1) ある固定された時点での諸演算が便利でない。(FORTRAN での DO による繰り返し演算 etc.)
 - 2) DYNAMO の性質上やむを得ないが、結果の numerical なエラーの検討が容易でない。
 - 3) 特殊な関数、特に判定に関する関数にもう一つ工夫が欲しい。
- などの点に改善の余地があるように思われる。

プログラムの原理

Industrial Dynamics (以下 ID と呼ぶ) は構成要素が因になり果となっている複雑な組織をフィードバック理論、意志決定機構の単純化を通して簡易に計算機システムにのせるように纏め上げた一つの思想である。

ID では次の六つの流れを組織内に考える。

- 1) 物 (Material) の流れ \longrightarrow
- 2) 金 (Money) の流れ \longrightarrow \$ \$ \$
- 3) 労働 (Personnel) の流れ \Longrightarrow
- 4) 注文、指示 (Order) の流れ \longrightarrow ○ ○ ○ ○ ○
- 5) 設備 (Capital equipment) の流れ \longrightarrow
- 6) 情報 (Information) の流れ \dashrightarrow

情報は上の五つの実体の流れから取り出され、システムの決定機構に作用する。

これらの流れは全て連続量として扱われ (すなわち一次の微係数を持つとする) 時間的変化をレート (rate) で考える。その流れの中で集積されたものをレベル (level) と呼ぶ。

たとえば河にダムがあったとすれば、1分当たり何トンの水が流れ込みまた放出するかはレートであり、一方ダムに貯えられる貯蔵水量がレベルに相当する。式で表わせれば 1 階の微分方程式の形で

$$\frac{dy}{dt} = x_{in} - x_{out}$$

y : レベル (貯水量)

x_{in} : レイト (流れ込む速度)

x_{out} : レイト (放出する速度)

となる。

これを計算機にのせるために差分方程式に変換して

$$y(T+\Delta T) = y(T) + \Delta T(x(T)_{in} - x(T)_{out})$$

とする。これが ID でいう レベルの基本的な式である。

DYNAMO 流に書けば J が 1 DT 時点前、K が現時点、L が 1 DT 時点後、JK, KL をそれぞれ時間間隔とすれば

$$Y.K = Y.J + DT(XIN.JK - XOUT.JK)$$

となる。

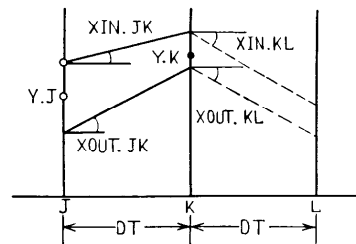
レイトの式としてたとえばダムの放出速度をその時点での貯水量に比例させるとすると

$$x(T)_{out} = y(T)/Const.$$

となり DYNAMO 流に表わせれば

$$XOUT.KL = (Y.K)/Const$$

である。



第 1 図 DYNAMO の時点表示

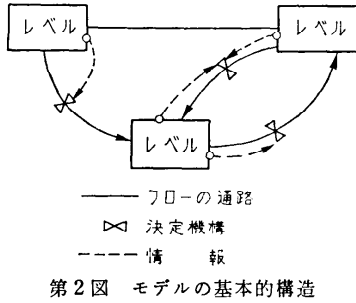
これがレートによる決定機構といわれるもので、組織の流れにおいてバルブのような動きをする。

すなわち複雑な産業や経済のモデルも分解すれば

- 1) 多数のレベル
 - 2) 一つのレベルの中味を他のレベルに運ぶためのフロー
 - 3) レベル間のフローの大きさ (すなわちレート) を決める決定機構
 - 4) 決定機構とレベルを結ぶ情報の通路
- の四つの factor のどれかに当てはまるとする。

この構造の把握の上に立ち DYNAMO で許される方式の形式を使ってモデルを組むことになる。

方程式の種類は上に述べたレベル、レイトの他に補助 (auxiliary), 補促 (supplementary), 初期値



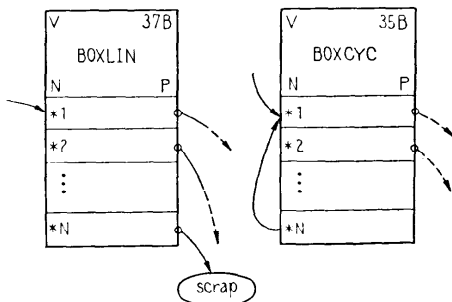
(initial value) の各方程式がある。

DYNAMO では方程式の形式が特殊の関数も含めて約 60 に制限されているので、作った方程式がその形式に該当しない場合は補助方程式を介在させて組み立てる。

補促方程式は output の際、必要なデータを作り上げるための方程式であり、初期値方程式は差分方程式を解くために当然必要な初期値を与える。

DYNAMO はこうして作られた組織を記述する全ての方程式を次の順序に従って計算する。

- 1) J と K の間のレベルの変化を元にして K におけるレベルの値を計算する。(すなわち J 時点のレベルと JK 時間間隔のレートから K 時点のレベルの値を計算する)
- 2) 時間を DT だけを K に進める。(TIME. J に DT を加え TIME. K とする)
- 3) 補助方程式の左辺の値が計算される。
- 4) JK 時間間隔のレートと K 時点のレベルから KL 時間間隔のレートを計算する。
- 5) output すべき変数の値を計算しテープに書き込む。
- 6) 時刻が予め決められた時刻になっていれば boxcar train をシフトする。



第3図 2種類の boxcar train

7) K, L を J, K にシフトする。

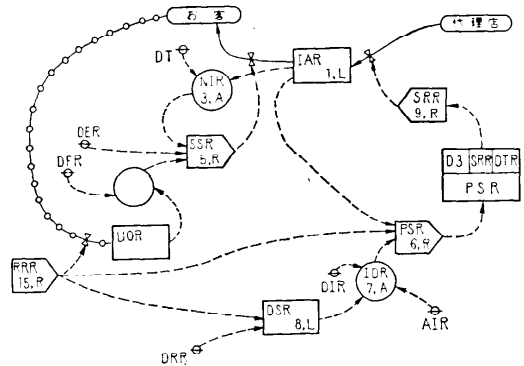
DYNAMO では現時点とその前後の時点の情報しか得られないので、さらに以前の情報なり、同じ情報がある期間ごとに週期的に引き出す(季節変動 etc.) ために boxcar train と呼ばれる temporary memory が用意されている。

小売店のモデル化の具体例

小売店が注文を受けると、この注文は製品の出荷まで受注残の中に入っている。製品の出荷は在庫が十分であれば受注残に比例する。

小売店は出荷量を補充し、在庫を最適水準に保つために代理店へ注文をだす。最適在庫水準は平均売上げの何週間分というようにきめる。

小売りが注文をだしてからある遅れがあって入荷する(第4図)。



第4図 ID によるフロー・ダイアグラム

[方程式]

(1) IAR (レベル: 小売店の在庫量) の方程式

$$IAR.K = IAR.J + (DT)(SRR.JK - SSR.JK) \quad (1.L)$$

IAR: 小売店の在庫量 (Units)
SRR: 代理店から小売店への入荷 (Units/Week)

SSR: 小売店からお客への出荷 (Units/Week)

(2) UOR (レベル: 受注残) の方程式

$$UOR.K = UOR.J + (DT)(RRR.JK - SSR.JK) \quad (2.L)$$

UOR: 小売店の受注残

RRR: お客から小売店への注文 (Units/Week)

(3) SSR (レート: 小売店→お客への出荷) の方程式

在庫が十分にあるときは、SSR. KL=UOR. K/DER
である。

もし (SSR. KL)(DT)>IAR. K であると時点Lで
IAR は負になってしまう。したがってこのときは

$$SSR. KL=IAR. K/DT$$

を採用しなくてはならない。すなわち

$$SSR. KL=MIN(UOR. K/DER, IAR. K/DT) \quad (3. L)$$

DER: 小売店が注文に応ずるおくれ (Weeks)

(4) PSR (レイト: 小売店から
代理店への発注) } の方程式
IDR (レベル: 最適在庫水準)

$$PSR. KL=RRR. JK+(1/DIR)(DIR. K - IAR. K) \quad (4. L)$$

PSR: 小売店から代理店への発注
(Units/Week)

DIR: 小売店の在庫調整のおくれ (Weeks)

IDR: 最適在庫水準 (Units)

$$IDR. K=(AIR)(RSR. K) \quad (5. L)$$

(5) RSR (レベル: USER からの注文の平均)

$$RSR. K=RSR. J + (DT/DRR)(RRR. JK-RSR. J)$$

これは指数平滑法 (exponential smoothing) を用

いた式である。

すなわち、単なる平均ではなく、現時点に近いもの
に大きな比重をかけた加重平均の一種である。

DRR: 指数平滑法を適用する期間 (Weeks)

(6) SRR (レイト: 代理点から小売店への入荷)

$$SRR. KL=DELAY 3(PSR. JK, DTR) \quad (6. L)$$

DTR: 遅れに対する定数 (Weeks)

[out put]

アウト・プットのためのコントロール・カードとして
PRINT, PLOT, SPEC の3種類のカードがある。

アウト・プットとして時系列的に表の形で、また図
形で得られるのが DYNAMO の特徴の一つであるが
PRINT では表に打ち出す変数とその様式 PLOT で
は図の形で打ち出す変数とその様式とSPEC で
はシミュレーションの時間刻みである DT の値、シ
ミュレーションの長さ、表に打ち出した図の形で打
ち出す時間間隔 (DT の整数倍の形で与える) を指定
する。

また DYNAMO では rerun カードを備えること
により、シミュレーションの構造を変えないで、定数
のみを変化させて次々と run を行なわせることがで
きる。

(昭和39年11月15日受付)

```

* SPUL-MCG,DYN,A00000001 000 A00001
RUN 2698JP
NOTE MODEL OF RETAIL STORE
NOTE
IL IAR.K=IAR.J+(DT)(SRR.JK-SSR.JK) INVENTORY ACTUAL
IL UOR.K=UOR.J+(DT)(RRR.JK-SSR.JK) UNFILLED ORDERS
20A NIR.K=IAR.K/DT NEGATIVE INVENTORY
20A STR.K=UOR.K/DFR SHIPMENTS TRIED
54R SSR.KL=MIN(STR.K,NIR.K) SHIPMENTS SENT
40R PSR.KL=RRR.JK+(1/DIR)(IDR.K-IAR.K) PURCHASE ORDERS SENT
12A IDR.K=(AIR)(RSR.K) INVENTORY DESIRED
3L RSR.K=RSR.J+(DT)(1/DRR)(RRR.JK-RSR.J) REQUISITIONS SMOOTHED
39R SRR.KL=DELAYS(PSR.JK,DTR) SHIPMENTS RECEIVED
NOTE
NOTE INITIAL CONDITIONS
NOTE
12N UOR=(DFR)(RRR)
6N RSR=PSR
12N IAR=(AIR)(RRR)
6N RRR=RRR
6N PSR=RRR
NOTE
NOTE INPUT
NOTE
7R RRR.KL=RRR+RCR.K REQUISITIONS RECEIVED
45A RCR.K=STEP(STH,5) REQUISITION CHANGE
NOTE
NOTE CONSTANTS
NOTE
C AIR=8 WKS CONSTANT FOR INVENTORY
C DFR=1 WK DELAY IN FILLING ORDERS
C DIR=4 WKS DLY REFILLING INVENTORY
C DRR=8 WKS REQUISITION SMOOTHED
C DTR=2 WKS DELAY IN TRANSIT
C RRI=1000 ITEMS/WK
C STH=100 ITEMS/WK STEP HEIGHT
NOTE
PRINT 1) IAR, IDR(2) UOR(3) RRR, SSR(4) PSR, SRR
PLOT IAR=1, UOR=0/RRR=R, SSR=S, PSR=S, SRR=Q
SPEC DT=0.5/LENGTH=50/PRTFRR=2/PLTFRR=0.5

```

第5図 モデルのリスト

TIME	IAR IDR	UOR	RRR SSR	PSR SRR
E+00	E+00	E+00	E+00	E+000 E+000
0.000	8000.0	1000.0	1000.0	1000.0
2.000	8000.0	1000.0	1000.0	1000.0
4.000	8000.0	1000.0	1000.0	1000.0
6.000	7968.0	1063.1	1100.0	1131.0
8.000	8094.6		1045.1	1015.0
10.000	7929.0	1095.8	1100.0	1180.0
12.000	8251.5		1095.8	1118.0
14.000	8344.6	1100.0	1099.5	1173.0
16.000	8342.1		1100.0	1168.0
18.000	8463.5	1100.0	1100.0	1134.0
20.000	8599.5		1100.0	1149.0
22.000	8548.0	1100.0	1100.0	1124.0
	8644.1		1100.0	1134.0
	8607.1	1100.0	1100.0	1117.0
	8678.8		1100.0	1124.0
	8649.8	1100.0	1100.0	1114.0
	8705.7		1100.0	1118.0

第6図 アウト・プット (プリント)

