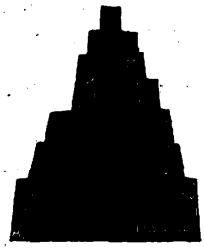

22. シミュレーション言語†

中西　俊男††

1. まえがき

シミュレーション言語（システム・シミュレータ，もしくはシミュレーション・プログラミング言語とも呼ばれる）は大別して，連続系，離散系および両者を統合した両系（combined system）に分類される．連続系のシミュレータは，もともとアナログ・コンピュータの代替として考えられたもので，Selfridge のディジタル・アナログ・シミュレータ（1955年）に始まってCSMP Ⅲ，CSSL Ⅲなどに至る25年の歴史を持っている．離散系のシミュレーション言語の嚆矢はGPSS，SIMSCRIPT（いずれも1961年）で，これらに続いて，SOL，SIMULA，SIMPL/I など数多くのシミュレーション言語が開発された．GPSS，SIMSCRIPT いずれも数段階に亘るバージョン・アップで，現在のシステムは当初の面影を残さない程の発展を見せている．離散系シミュレーション言語 GASP が進化した GASP Ⅳ は，両系シミュレーション言語の代表で1970年代に入ってお目見えした．

これら数多くのシミュレーション言語は，モデル構成の考え方（world-view）や言語形式にそれぞれ特徴を有しており，これがプログラミングの難易，モデル構成の融通性に深く関与している．ここでは，離散系シミュレーション言語を中心に，連続系，両系シミュレーション言語の新しい傾向，特徴についてふれてみる．

2. 離散系シミュレーション言語[1)~4)]

離散系シミュレーション言語の特質を論ずるとき，その言語でどのような

（1）静的モデル構成概念

（2）動的モデル構成概念

が取り入れられているかを考察する必要がある．前者は，システムを記述するデータ構造の考え方であり，後者はシミュレーション時間の経過に伴って生起するシステムの変化を記述する際の考え方である．データベース・マネージメント・システムでは，前者に重点を置き，後者に関しては，実時間に伴って見られるデータの変化を追跡するに過ぎない．シミュレーション言語では，後者が主体であり，単にデータのみならず，システムのあらゆる変化をシミュレーション時間（実時間のモデル．実時間そのものが使われるものもあり，リアルタイム・シミュレーションと呼ばれる）に対応させて追跡する．ただ最近の傾向としては，シミュレーションの静的モデル構成機能の不備が指摘され，システムを記述するデータ構造の充実を標榜するシミュレーション言語もいくつか見られる（SIMSCRIPT Ⅱ.5[4),6)~8)]，SIMULA[9),10)]等）．

モデル構成概念は，シミュレーション言語により種種様々であるが，大別してきわめて包括的な概念と，個別的もしくは分割的概念からなるものとが見られる．通常，包括的概念の方が融通性があり，高級で優れていると考えられる傾向があるが，度が過ぎると難解に過ぎ，却って不便になるきらいがある．たとえば，SIMULA の class という概念は，あまりにも包括的で，極端な言い方をすれば，何でも class で，初心者には（のみならず相当のベテランでも）きわめて理解しにくい概念である．包括性のメジャでいえば，GPSS，SIMSCRIPT，SIMULA の順に高度化する．

シミュレーション言語には，シミュレーション専用の言語（たとえば GPSS）と，シミュレーション機能を兼ね持つ汎用プログラミング言語もしくはシステム記述言語の形態のもの（たとえばSIMSCRIPT Ⅱ. 5，SIMULA 等）がある．後者は，もともと汎用プログラミング言語，FORTRAN，ALGOL 等をホスト言語とし，これに抱かれる形で開発されたため，その汎用化はきわめて容易であったと見られる．

離散系シミュレーション言語の古典的代表格は，GPSS と SIMSCRIPT である．当初この二つの言語の動的モデル構成概念が，システム・シミュレータの

† Simulation Programming Languages by Toshio NAKANISHI (The University of Seikei).

†† 成蹊大学工学部

基本的 world-view としてあげられた．すなわちトランザクション中心 (transaction-oriented) の world-view と，事象中心 (event-oriented) の world-view である．前者はトランザクション (時間の経過と共にシステムの中を動く対象) の流れる路を構成すれば (したがって各命令は路の構成要素にほかならない) シミュレーション・プログラムになるという，きわめて具体性に富む考え方である．後者はシステムに変化をもたらす事象を，各時点 (時間経過はない) ごとに記述するルーチンの集合でシミュレーション・モデルを構成する方式である．この両者にやや遅れて，プロセス中心 (process-oriented) の world-view が登場した．ここでいう process とは，SIMSCRIPT II. 5 を開発した CACI によれば，"a sequence of related events separated by predetermined or indefinite lapses of time." ということである．この意味では，GPSS によるモデルも明らかに process-oriented であり，event-oriented な world-view も，少し調整すれば，process-oriented になるわけで本質的な差異は認め難い．SIMSCRIPT もいつの間にか event 中心から process 中心に移りかわっている．最近の新しいシミュレーション言語もしくは，新しいバージョンのフィロソフィは，おしなべて process-oriented world-view である．

2.1 GPSS[2),3),5)]

モデル構成概念が最も具体的で，プログラミングのやさしい実用的シミュレーション言語である．GPSS, GPSS II, 同III, 同 360, 同IVを経，GPSS Vが最も新しいバージョンであるが，コンピュータ・メーカ各社それぞれにVレベルの GPSS (名称はそれぞれに異なるが) を開発している．古くから，待行列型のモデルに向くシミュレータとされてきたが，ほかのシミュレータの持つ機能を付加して，汎用性を上げてきた．user chain, group entity がその例である．ただインタプリタ形式の言語のため，オンライン化等は比較的容易ではあるが，計算機能，データ構造の記述機能では劣っており，特にトポロジカルなシステムの記述に難点がある．この点を誇る SIMULA などから格好の餌食にされているようであるが[13),16)]，SIMULA 側の消化不良がないとも言えず全般的な白黒はつけ難い．facility, storage 等システムの恒久要素や待行列などに関する統計結果がシミュレーション終了時点 (もしくは途中) で自動的に出力される点や，モデルの validity のチェック機能が優れているなど，捨て難い味を持つ．オンライン機能，特にグラフィック機能の付加等についても多くの試みがなされ (GPSS/NORDEN のグラフィック特性)，中には，手書きもしくはメニュを拾ってつくられるフロー・チャートからシミュレーション・プログラムを構成できるものもある (Grail system, RAND tablet, OLSS I[18)])．ほかのシミュレーション言語から，度々攻撃目標にされるようであるが，それだけの強みと実用性を持つシミュレーション言語と見られる．

2.2 SIMSCRIPT

RAND Corporation で開発された SIMSCRIPT に始まり，SIMSCRIPT I. 5, 同II, 同II. 5 とバージョン・アップされて現在に至っている．ここでは最新バージョンの SIMSCRIPT II. 5[6)~8)] について簡単に紹介する．

SIMSCRIPT II. 5 は，計画中のものを含めると，level 1 から level 7 に至る綜合的なシステム記述言語である．level 1～level 3 で，FORTRAN 相当の機能をカバーしている．level 4 で初めて，静的モデル構成概念である permanent entity, temporary entity, set およびそれぞれの attribute が取り扱われる．level 5 が加わって動的モデル構成概念すなわちシミュレーション機能が備わり，タイミングのコントロール，確率過程の表現等が可能になる．

level 6, 7 は，計画中 (あるいは部分的に完成している) のもので，level 6 はデータベース機能の整備，level 7 は，いわゆる "language writing language" としての機能を目指している．なお，最近では，連続変数を取り扱える機能の追加も考えられており，GASP IV と同様，両系のシミュレーション言語に発展すると見られる．

SIMSCRIPT II. 5 での最近の進展は，事象中心の考え方から，プロセス中心の考え方に移ったことである．すなわち，以前のバージョンでの event routine では，ある1時点での状態の変化しか考えられなかったが，II. 5 の process routine では，いくつかの異なる時点での状態変化を記述できるようになったのである．プロセスの取り扱いのための命令 (suspend, reactivate および interrupt, resume 等．前者はプロセスを一時停止もしくは待ち状態に留め，ある条件が満たされたとき再始動させる命令であり，後者はほかのプロセスの割り込みとその解除を行う命令である．GPSS の queue, depart, link unlink, preempt, return などに対応するがある種のモデルでは，SIMSCRIPT

のそれの方が融通性が高い.) も付加されている.

SIMSCRIPT II.5 では, このほか entity, set, event notice, その attribute の定義, 統計計算の指定などを, 特殊の定義フォーム (SIMSCRIPT Definition Form) に依らず, PREAMBLE と呼ばれるあるセクションで, 記述的に行う形になっている. このように, ある意味で自由度は増え, 一見プログラムが理解しやすくなっているようであるが, 規格化と反対の方向に向うことは進歩といえるかどうかは議論の余地がある.

SIMSCRIPT II.5 といってもすべて同じ言語仕様とは限らない. 数多くの解説書が出ているが, 許される命令数そのほかかなり差異があるので注意を要する.

2.3 SIMULA

SIMULA は, 1962 年 K. Nigaard により離散型事象ネットワークの記述のために, ALGOL を拡張して作られたもので, 現在使われているバージョンは, シミュレーション機能を備えた汎用プログラミング言語というべきものである. 本特集では別に SIMULA の解説があるので, ここではシミュレーション言語としての特徴について簡単に触れるにとどめる.

SIMULA のシミュレーション機能は, モデル構成概念 "class" に集約される. class の中に, システムの静的動的記述をすべて含めている. すなわちシステムを記述する entity, set やその変数, 更にプロセス・ルーチン (ただしこれは process class と呼んでいる) をすべて class と呼ぶわけである. システムに組み込まれた system class というものもあり, たとえば待行列などは system class SIMSET を用いて記述するが, この時更に link class, head class というものを宣言する必要がある. 一口では説明できないが, SIMULA は, システムを記述するデータ構造に特別の配慮をしてあり, GPSS では簡単には記述できない, トポロジカルな関係も容易に表現できる特性がある. しかし一方 GPSS などで標準化されたモデル (block 命令) を SIMULA で表現しようとすると相当に複雑難解になる[16]. 結局, シミュレーション言語一般に言えることは, 融通性をある程度犠牲にし, 標準化基本モデルをベースにしてプログラミングを容易にするか, プログラミングは難しくなるが汎用性, 柔軟性を高める方向を打ち出すか, の二者択一の問題ということである.

2.4 SIMPL/I[12]

SIMPL/I (Simulation Language Based on PL/I) は, IBM のプロジェクト・チームによって, 1968～1970 年に開発された離散型シミュレーション言語で, process oriented な world-view をもっている.

PL/I をホスト言語としているために, 汎用性はきわめて高く, データ構造の記述, レポート機能, I/O 処理など, PL/I の特性がそのまま持ち込まれている. 反面, PL/I の難解さそのものも受けついでおり, 日本であまり使用されていないという点では SIMSCRIPT と似ている.

一方, ほかのシミュレーション言語が各社の計算機に開発されているのに反し, SIMPL/I は IBM のみの市場であり, その動向が注目されている.

SIMPL/I におけるモデルの表現は, process, entity および list なる概念で行われる. いわゆるシミュレーション機能は process と list 概念である.

特徴的なことといえば, SIMSCRIPT の set はその構成要素を entity のみに限定されるが, SIMPL/I では, list 自身をもその構成要素として list を構成することができる点であろう. これによってモデル構成の融通性が一段と増し, 高度なモデル化を可能としているといえる.

以上のほか, SOL, GASP そのほか数多くの離散系シミュレーション言語が開発されているが本解説では紙面の都合上割愛せざるを得なかった.

3. 連続系シミュレーション言語

初期の連続系シミュレーション言語は, 良きにつけ悪しきにつけ, 往時のアナログ・コンピュータの機能をそのまま模型化したものであったが, 1964 年頃から, 従来のアナログなブロック・モデリングの機能にディジタルなプログラミング機能, すなわち FORTRAN のような代数的, 手続的機能が加えられるようになり, 融通性, 汎用性に格段の進展を見せた. 1965 年 Syn と Wyman により開発された DSL/90 は, この考え方にもとづいてつくられた第1号の FORTRAN ベースのシミュレータである. この頃から従前の Digital Analog Simulator (DAS) の呼称が, 連続系シミュレータ (Continuous System Simulator) に変わる. 現在最も進んだ形のシミュレータの代表として, CSMP III, CSSL III などがある.

アナログ・コンピュータの特徴は, 並行処理機の特性でもってきわめて高速であること, マン・マシーン・インタラクションが容易であることとされている

が，連続系シミュレータでも，この方向に向けての努力が続けられている．PACTOLUS[25]から始まるシミュレータのオンライン化[23]と，マルチ・プロセッサ方式によるシミュレーションのリアル・タイム化である．後者に関しては，そのための専用機[22),24)]も数多く開発されており，リアル・タイム・シミュレーションにもとづく制御に利用される傾向が見られる．

連続系シミュレーション言語の一つに DYNAMO がある．DYNAMO は，MIT の Forrester, J. W. によって考え出された Industrial dynamics に基づくモデルを構成するためのもので，社会経済モデル等マクロ・モデルに適用される．連続で，閉じたループを構成する情報フィードバック・システムのシミュレーションを行ったためのシミュレーション言語である．インダストリアル・ダイナミクスからアーバン・ダイナミクス，システム・ダイナミクスあるいはワールド・ダイナミクスに発展している．言語は DYNAMO に始まり，現在 DYNAMO Ⅱが利用できる．

4. 両系シミュレーション言語

両系のシミュレーションとは，簡単に言って，連続系と離散系を同時に扱うシミュレーションということである．連続系では，シミュレーション・クロックが微小等時間隔で刻まれ，離散系では不等時間隔で生起するキイ・イベントで刻まれる．不等時に生起する事象には，スケジュールされた時刻で起るもの (time event) とある状態変数がある条件を満たした時起るもの (state event) があるが，両系のシミュレーションでは，等時間隔の時刻と不等時間隔の時刻が重なって刻まれるわけである．両系のシミュレーションでは，これらいくつかの時刻の調整が重要なポイントになる．

連続系シミュレーションはそもそもモデルの精細な記述を，離散系のそれは細部を省略した大局的記述を行うわけで，両系ではその意味でのモデル表現のアンバランスは避けられない．つまり連続系でとらえたせっかくの精度か，離散系で無駄にされる危険がある．その意味で，シミュレーションの対象の選択に留意する必要がある．

両系のシミュレータとして最も有名なのは，GASP Ⅳである．GPS Ⅳは FORTRAN ベースのシミュレータで，FORTRAN で書かれた GASP ルーチンを使ってモデルを構成する．静的モデル構成のための機能も特別に準備されている．最近では，FORTRAN の代りに PL/I を使った GASP—PL/I も開発されており，またディスプレイ出力の工夫が盛られた GASP Ⅳ/E というバージョンもある．

チューリッヒ工科大学では，GASP Ⅴを開発中といわれるし，SIMSCRIPT Ⅱ.5の両系化も進んでいる．今後利用分野の拡大に伴い，それぞれの応用にマッチした専用の両系シミュレータの開発が期待される．

あとがき

紙面の都合上，各種シミュレーション言語の特性を十分に解説できなかった．関心をお持ちの方は参考文献を見ていただきたい．特に文献4）が良い参考文献である．

参考文献

1) Buxton, J. N.: Simulation Programming Languages, p. 463, North-Holland publishing Company, Amsterdam (1968).
2) Gordon, Geoffrey: Systems Simulation, p. 324, Prentice Hall, Englewood cliffs, New Jersey (1978).
3) 中西俊男：コンピュータ・シミュレーション p. 276, 近代科学社 (1977).
4) Nabil R. Adam and Ali Dogramaci: Current Issues in Computer Simulation, p. 297, Academic Press, New York (1979).
5) Bobillier, P. A. et al.: Simulation with GPSS and GPSS V, Prentice Hall, Englewood Cliffs New Jersey (1976).
6) Kiviat, P. J. et al.: SIMSCRIPT Ⅱ.5 Programming Language, p. 384, CACI (1975).
7) SIMSCRIPT Ⅱ.5 Reference Handbook, p. 251, CACI (1976).
8) A Quick Look at SIMSCRIPT Ⅱ.5 p. 35, CACI (1979).
9) Birtwistle, G. M. et al.: SIMULA Begin, Auerback Pub., Inc., Philadelphia (1977).
10) Burroughs B 5500 Information Processing System SIMULA Reference Manual, Burroughs Corp. (1970).
11) A. Alan and B. Pritsker: The GASP Ⅳ Simulation Language, p. 451, John Wiley & Sons, New York (1974).
12) SIMPL/I Program Reference Manual, p. 205, IBM (1972).
13) Atkins, Stella: A Comparison of SIMULA and GPSS for Simulating Sparse Traffic, Simulation Vol. 34, No. 3, pp. 93-100 (1980).
14) Annino, J. S. and Russel, E. C.: Successful Simulation of Complex System with SIMSCRIPT Ⅱ.5, Simuletter Vol. 10, No. 1, 2, pp. 7-14 (1979).

15) Schmidt, Bernd: The Way ahead in Discrete Simulation, Proc. 1980 SCSC pp. 46-49 (1980).
16) Houle, Philip A.: On the Structural Concept of Simula and Simulation Modeling, Proc. 1974 SCSC, pp. 55-60 (1974).
17) Eklundh, Berth: SIMULA-A Way of Thinking, Proc. 1979 WSC. pp. 11-20 (1979).
18) 吉沢節子, 中西俊男: オンライン・シミュレーション・システム OLSS-1 の開発, 成蹊大学工学報告, No. 20, pp. 1465-1484 (1975).
19) Henriksen, James O.: An Interactive Debugging Facility for GPSS, Proc. 1977 WCC, pp. 331-338 (1977).
20) CSMP Ⅲ Program Reference Manual, IBM.
21) CSSL Ⅲ User's Guide, CDC.
22) System 10 Dynamic Simulation System, ADI Manual.
23) Benham, R. D.: Interactive Simulation Language-8 (ISL-8) Simulation, Vol. 16, No. 3, pp. 116-129 (1971).
24) Korn, Granino A.: Back to Parallel Computation: Proposal for a Completely New on-line Simulation System Using Standard Minicomputers for Low-cost Multiprocessing. Simulation Vol. 19, No. 2, pp. 37-45 (1972).
25) 中西俊男: オンライン・シミュレーション: 情報処理, Vol. 8, No. 6, pp. 345-354 (1967).

(昭和56年2月2日受付)