

# Programe de aplicații

## Sistemul de Simulare SOL/PC

mat. Viorica Madan, mat. Maia Ungureanu

Institutul de Matematică al Academiei de Științe  
Chișinău, Moldova

### 1. Introducere

Simularea este utilizată pe o scară largă în diverse domenii la cercetarea sistemelor complexe. Conform opiniei unor experți în informatică, se poate afirma că ultimul deceniu al secolului 20 va fi deceniul limbajelor de simulare. Unul dintre cele mai compacte limbaje este SOL (Simulation Oriented Language) propus de Knuth și McNeley [1]. Întrucât în prezent calculatoarele personale sînt foarte răspîndite, limbajul SOL se poate implementa pe acestea.

De asemenea, este extrem de actuală și construirea programelor de calitate. Calitatea produsului-program include caracteristici precum: eficacitatea, veridicitatea, dimensiunea, complexitatea, costul elaborării, timpul de executare etc. Multe din aceste caracteristici pot fi determinate în mod automat din textele programelor. Procesul de determinare și estimare a calității programelor oferă posibilitatea de a dirija calitatea acestora.

Sistemul propus în acest articol include atât elementele clasice ale unui sistem integrat (menu multiwindow, redactor pentru texte, compilator etc), cit și elementele de estimare a calității modelelor și de reprezentare grafică a rezultatelor simulării.

### 2. Descrierea sumară a limbajului SOL

SOL este un limbaj de tip ALGOL, care reflectă comod concepții de bază ale modelării ca: timpul de simulare, dezvoltarea paralelă a proceselor și evenimentelor în acest timp, independența și corelația lor relativă, cozi (stocuri) și dispozitive.

Componentele globale ale SOL-ului sînt dispozitivele, cozile și procesele. Mai mult decît atât, limbajul permite folosirea în expresii aritmetice a variabilelor aleatoare (tip random) cu diferite legi de repartiție și de a obține informația statistică standard despre model la sfîrșitul simulării (de ex. coeficientul de utilizare a dispozitivelor și cozilor, volumul maximal ocupat din stoc, tabele, histograme, diagrame și altele).

Modelele scrise în SOL sînt laconice, permit observarea structurii și interacțiunii dintre elementele lor și pot fi ușor modificate.

Programul SOL constă din descrierea componentelor globale (operatorii INTEGER, REAL, FACILITY, STORES, MONITOR, TABLE) și proceselor (operatorul PROCESS). Descrierea proceselor conține o consecutivitate de diferiți operatori, ca

într-un limbaj de programare tradițional.

Paralelismul modelului este susținut de un șir de operatori speciali ca:

**NEWTRANSACTIONTO** - începutul (crearea) unui nou eveniment;

**READTO** - începutul (crearea) unui nou eveniment cu citirea valorilor pentru datele lui locale dintr-un fișier extern;

**CANCEL** - sfîrșitul (lichidarea) evenimentului.

În SOL este folosită metoda ceasului variabil, timpul de simulare fiind indicat în unități discrete în operatorul WAIT. Operatorul STOP oprește simularea și pregătește rezumatele statistice.

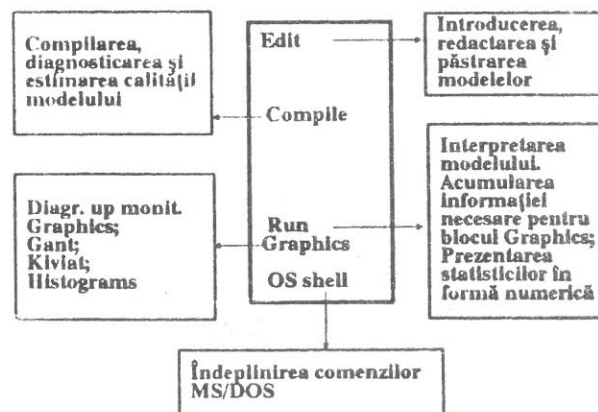
Pentru a facilita experimentarea modelului în sistemul SOL/PC este introdus operatorul EXPERIMENT. Se înțelege prin model un program scris în limbajul SOL.

### 3. Structura funcțională a sistemului SOL/PC

Sistemul SOL/PC îndeplinește următoarele funcții:

1. Redactarea textului modelului;
2. Compilarea modelului;
3. Interpretarea operatorilor de bază din SOL (ca RANDOM, NORMAL, EXPONENTIAL, GEOMETRIC, POISSON etc) și simularea cvasiparalelismului modelului;
4. Output al informației statistice privind rezultatele experimentării:
  - în formă numerică;
  - în formă grafică (diagramele Gant și Kiviat [2], diagramele după monitor, histograme);
5. Output al vectorului calității modelului;
6. Evaluarea multicriterială a calității modelului.

#### Schema sistemului integrat SOL/PC



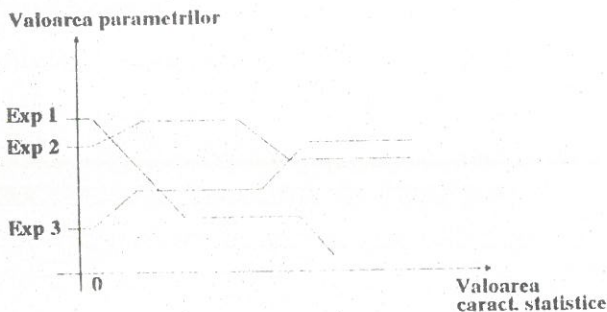
Ne vom opri mai detaliat la punctele 4-6.

#### 4. Interpretarea grafică a rezultatelor simulării

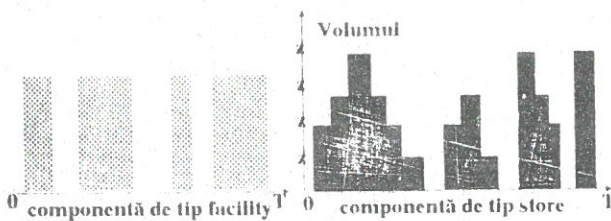
Pentru a se facilita interpretarea și analiza rezultatelor simulării, în sistem a fost introdus un pachet de programe de reprezentare grafică a statisticilor rezultante. Toată informația statistică ce rezultă din executarea modelului este acumulată în formă compactă într-un fișier special și servește drept date input pentru acest pachet.

În baza informației acumulate se construiesc următoarele grafice:

- 1) Graficul arată dependența caracteristicilor statistice ale obiectelor de variația valorii unui parametru din model, de la un experiment la altul. Valoarea parametrului este citită prin intermediul operatorului **EXPERIMENT**:



- 2) Diagramele după monitor reflectă durata de ocupare și eliberare a dispozitivelor (tip **facility**) și stocurilor (tip **store**). În cazul al doilea se arată câte unități din stoc au fost folosite.

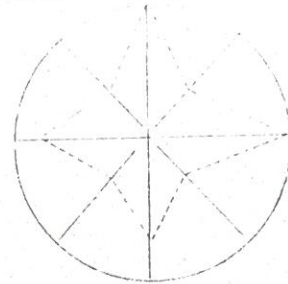


- 3) Diagramele Gant prezintă așa numitele profiluri de utilizare a obiectelor modelului. Ele permit experimentatorului să vadă în ansamblu, în formă grafică, stările de activitate ale diferitelor componente din sistemul modelat și suprapunerea lor în timp:

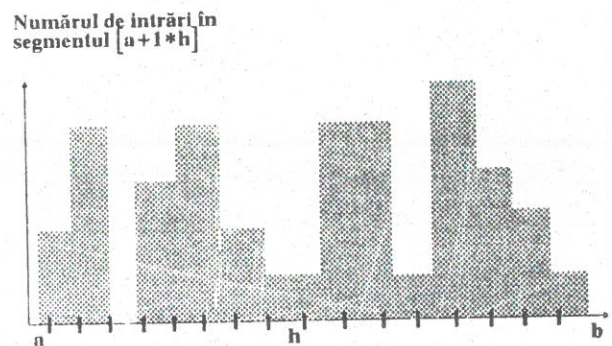


- 4) Diagramele Kiviat reprezintă un grafic încercuit. Razele cercului formează axele graficului. Pe aceste raze sînt depuse consecutiv valorile

parametrilor cu influență "pozitivă" și "negativă" asupra sistemului modelat. În final se obține o stea. Cu cît steaua e mai regulată, cu atît sistemul simulat e mai echilibrat:



- 5) Histogramele ilustrează în ce mod au fost completate tabelele (descrise cu ajutorul operatorului **tabulate**) în timpul executării modelului:



**a, b - granițele tabelului, h - pasul**

Toate diagramele acestea sînt obținute de experimentator prin intermediul unui dialog interactiv cu sistemul SOL/PC, în care se precizează numele parametrilor și caracteristicilor statistice ale obiectelor descrise în model.

#### 5. Estimarea calității modelului

Complexitatea textului modelului constituie unul din factorii de bază ce caracterizează calitatea lui. Ea poate fi prezentată prin parametrii (dimensiunea, structura modelului. Dimensiunea include lungimea modelului, volumul de memorie ocupat (în biți), mărimea bazei de date. Iar structura include numărul de moduli din model, interfața lor, structura lui logică.

În dependență de textul fiecărui model, se pot obține următoarele caracteristici: **N** - lungimea modelului, **n** - vocabularul lui, **V** - volumul (în biți), **E** - mărimea efortului necesar pentru a codifica modelul, **D** - complexitatea (după Holsted [3]); **RBD** dimensiunea bazei de date, **v** - numărul ciclomatic (după [4]); **EPT** complexitatea logică totală; **EPM** - complexitatea logică maximală (după [5]).

Toată informația necesară pentru a primi aceste caracteristici este accesibilă compilatorului, deci e poate calcula vectorul complexității pentru model

Notăm acest vector prin

$P(N, n, V, E, D, RBD, v, EPT, EPM)$

Întrucât testarea și ajustarea modelului impune folosirea compilatorului de multiple ori, se poate obține o mulțime de vectori  $\{P_i\}$ ,  $i = 1, 2, 3, \dots$ . Această informație se înregistrează în baza informațională a sistemului și este folosită la analiza complexității modelului. Și anume, se efectuează analiza multicriterială pe mulțimea vectorilor  $\{P_i\}$ , unde coordonatele vectorilor servesc drept criterii. Fie mulțimea  $\{P_i\}$  care formează un spațiu metric  $M$  cu distanța  $\rho(P_i, P_j)$  (analogic distanței din spațiul  $L_1$ ) definită pentru fiecare pereche  $P_i, P_j \in M$ . Analiza mulțimii  $\{P_i\}$  se efectuează în concordanță cu principiul găsirii elementului minimal din mulțimea vectorilor. Pentru această mulțime se stabilește o ordonare parțială în mod obișnuit, adică, pentru  $P_i$  și  $P_j$  din  $M$ , are loc  $P_i < P_j$  atunci și numai atunci când  $P_{ik} < P_{jk}$ , pentru  $k=1, 2, \dots, 1, k \in J$ , unde  $J$  este mulțimea indicilor. Elementul minimal al acestei ordonări parțiale formează Minima mulțimii  $M$  [6].

În rezultatul analizei multicriteriale, sistemul propune indicile celei mai reușite versiuni a modelului.

În afară de aceasta, sistemul SOL/PC permite obținerea diverselor diagrame ale estimării calității modelului în timp.

Acest fel de abordare a fost aplicat la elaborarea

modelelor în limbajul de simulare SOL pentru calculatoarele IBM PC, ceea ce a permis îmbunătățirea calității modelelor-program. Sistemul este implementat prin TURBO C în sistemul operațional MS/DOS. Acest fapt permite transferul sistemului la orice tip de calculator, dacă la acesta este realizat standardul limbajului C.

## Bibliografie

1. KNUTH, D.E., McNELEY, I.L.: *The Formal Definition of SOL*. In: IEEE Trans. on Electr. Comp., 1964, EC-13, pp. 409-416.
2. FERRARI, D.: *Computer Systems Performance Evaluation*. Editura Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1978.
3. HALSTEAD, M.H.: *Elements of Software Science*. Purdue University, Editura Elsevier, New York, 1977.
4. McCABE, T.: *A Complexity Measure*. In: IEEE Trans. on Soft. Eng., decembrie, 1979, pp. 308-320.
5. CATER, S.C., ș.a.: *Computation of Logical Effort in High Level Languages*. In: Computer Languages vol. 9,4/4, 1984, pp. 133-149.
6. KUGH, H.T., LUCCIO, F., PREPARATA, F.P.: *On Finding the Maxima 0 Set of Vectors*. In: J. ACM 22,4 octombrie, 1975, pp. 469-476.