

Curs 8 –GPSS

Autor: lect.dr. Florentina Suter

- a. Scopul simulării este de a crea modele ale unor sisteme reale, care pot fi programate și prin intermediul cărora poate fi studiată evoluția în timp a sistemelor.
- b. Pentru a programa un model de simulare se pot folosi diverse limbaje de programare.
- c. Unul dintre limbajele specifice Simulării este GPSS-ul. Limbajele specifice simulării au ca avantaj existența unor entități deja implementate, care caracterizează un model de simulare. Un program în GPSS constă din descrierea unui sistem iar asupra acestui sistem se pot face experimente (simulări) prin rularea programului.
- d. De unde poate fi luat soft-ul GPSS și documentația: <http://www.minutemansoftware.com/>.

Un program în GPSS este format dintr-o înșiruire de instrucțiuni scrise de obicei pe o singură linie.

Partile unei instrucțiuni în GPSS se numesc câmpuri. Aceste câmpuri sunt: numărul instrucțiunii (optional), eticheta (de cele mai multe ori opțională), un verb care semnifică acțiunea care are loc, operanții, comentarii. Comentariile sunt precedate de “;”.

Entități importante pentru un model descris de un program în GPSS: blocuri, tranzacții, SNA, facilitati:

- Acțiunea care definește o instrucțiune este exprimată ori printr-un bloc, ori printr-o comandă. Blocul este elementul de bază al descrierii unui model în GPSS.
- Tranzacțiile sunt entitățile care dinamizează modelul descris de GPSS prin trecerea lor prin blocuri. Fiecărei tranzacții îi este asociat un număr. Tranzacțiile de cele mai multe ori sunt asociate cu clienții dintr-un sistem de așteptare modelat de un program în GPSS.
- SNA – System Numeric Attributes sunt entitățile care dau diferite informații numerice despre sistemul modelat de program în GPSS. Se folosesc în operanți sau expresii.
- Facilitățile sunt entități cu ajutorul cărora se simulează serviciul cu un singur punct de servire.

Primul program în GPSS!

Se deschide GPSS World, File, New, Model. În editorul care se deschide se scrie următorul program:

GENERATE 58
QUEUE COADA
SEIZE BENZINA
DEPART COADA
ADVANCE 51
RELEASE BENZINA
TERMINATE 1

Este vorba de simularea unui sistem de asteptare, in particular de functionarea unei benzinarii la care clientii vin la fiecare 58 unitati de timp (in GPSS nu conteaza care este unitatea de timp, conteaza sa fie aceeasi in tot programul), sunt inregistrati in coada, sunt preluati de facilitatea indicata (punctul de serviciu), se marcheaza iesirea lor din coada, primesc serviciul care dureaza 51 de unitati de timp, parasesc facilitatea si, prin ultima instructiune este marcata iesirea lor din sistemul asociat functionarii unei benzinarii.

Definitiiile blocurilor:

GENERATE

- creeaza tranzactii care vor parcurge modelul;
- are forma generala:

GENERATE A,B,C,D,E

unde

- A reprezinta timpul mediu intre generari. Daca operandul B nu exista, atunci generarile se vor face la intervale constante de timp, egale cu A;
- B poate fi un numar sau un nume de repartitie statistica. Daca este un numar, atunci generarile se vor face dupa o repartitie uniforma pe intervalul $[A-B, A+B]$. Adica timpul intre generari poate lua orice valoare intreaga din acest interval. Toate valorile au aceeasi probabilitate de a fi alese. Daca B este un nume de repartitie statistica, atunci timpii intre generari vor fi valori ale variabilei aleatoare cu repartitia B si media A.
- C este lungimea intervalului de timp cu care generarea tranzactiilor poate intarzia;
- D este limita de generare, numarul maxim de tranzactii care pot fi generate;
- E este prioritatea pe care o vor avea tranzactiile generate.

QUEUE si DEPART

- sunt doua blocuri care nu pot exista unul fara celalalt intr-un model de simulare. Ele nu sunt elemente ale modelului, ci aduna informatii despre asteptarea clientilor, adica informatii despre cozile care eventual se formeaza. QUEUE inregistreaza momentul la care un client intra in coada si DEPART momentul la care un client iese din coada.

- forma lor generala este

QUEUE A,B

DEPART A,B

unde

- A este numele cozii pentru care sunt adunate informatii;
- B este numarul de tranzactii care sunt inregistrate pentru intrarea/iesirea din coada. Daca lipseste este 1.

SEIZE si RELEASE

- sunt doua blocuri care nu pot exista unul fara celalalt intr-un model de simulare. Ele simuleaza intrarea unui client (tranzactie) la un serviciu (facilitate), respectiv iesirea unui client de la un serviciu (facilitate).
- forma lor generala este

SEIZE A

RELEASE A

unde A este numele sau numarul facilitatii.

ADVANCE

- este un bloc care simuleaza oprirea tranzactiilor in sistem pentru un anumit interval de timp. Oprirea poate fi determinata de diverse cauze. De exemplu aici oprirea tranzactiilor se face pentru simularea primirii unui serviciu.
- forma generala este

ADVANCE A,B

unde A si B au aceeasi semnificatie pe care o au operanzii A si B de la blocul GENERATE.

TERMINATE

- este un bloc care simuleaza iesirea clientilor din sistem, distrugand tranzactiile create de GENERATE.
- forma generala este

TERMINATE A

unde A reprezinta numarul care se scade din numarul total de tranzactii pentru care se ruleaza modelul.

- fiecarui bloc GENERATE trebuie sa-i corespunda un bloc TERMINATE.
- in orice model trebuie sa existe un bloc TERMINATE cu operandul A diferit de 0, pentru ca in acest mod se controleaza timpul de rulare. Rularea unui program in GPSS se face prin trecerea unui anumit numar de tranzactii prin sistemul modelat.

Se salveaza programul: File, Save.

Se compileaza: Command, Create simulation.

Se ruleaza: Command, START. Apare o fereastră în care este scris START 1. Se înlocuiește 1 cu o valoare întreaga pozitivă reprezentând numărul de tranzacții (clienți) pentru care se face experimentul de simulare.

Dupa rulare apare automat raportul creat in urma simulării. Dacă dorim o rulare fara afisarea automata a raportului se scrie in fereastra de start, de exemplu, START 100,NP.

La o privire rapida peste raport observam ca:

- este afisat timpul de inceput si timpul de sfarsit al simulării, numărul de blocuri din model, numărul de facilitati, numărul de entitati STORAGE.
- sunt afisate blocurile din model si in dreptul fiecaruia numărul de tranzactii care au trecut pe la acel bloc.
- sunt date informatii despre facilitatea BENZINA:
 - ENTRIES – cate tranzactii au trecut pe la aceasta facilitate (cati clienți au fost serviti)
 - UTIL. – ce fractiune (raportat la 1) din timpul total de simulare a fost folosita facilitatea
 - AVE. TIME – timpul mediu de servire.
 - AVAIL., OWNER, PEND, INTER, RETRY, DELAY au semnificatie pentru modele mai complicate si nu le discutăm acum.
- informatii despre coada COADA
 - MAX - lungimea maxima a cozii;
 - CONT – lungimea cozii la sfasitul simulării;
 - ENTRY – cate tranzactii au fost întregistrate in coada.
 - ENTRY(0) – cate tranzactii care nu au asteptat deloc au fost întregistrate in coada.
 - AVE. CONT – lungimea medie a cozii.
 - AVE. TIME – timpul mediu de asteptare.

- AVE.(-0) – timpul mediu de asteptare in coada pentru clientii care au asteptat un timp mai mare decat 0.
- Informatii despre agenda evenimentelor viitoare: FEC (Future Events Chain). Se presupune ca urmatorul eveniment va fi sosirea celui de-al 101-lea client la momentul 5858.

Ferestre care pot fi vizualizate dupa si in timpul simularii:

- a blocurilor: Window, Simulation window, Blocks window;
- a facilitatilor: Window, Simulation window, Facilities window;
- a cozilor: Window, Simulation window, Queues window;

Se mai face o rulare a programului cu una dintre aceste ferestre deschise si se observa modificarile care au loc in timpul rularii. Timpul de rulare este mai lung pentru ca este deschisa o fereastră grafica.

Pentru oprirea experimentului se tasteaza F4. Pentru continuarea lui F2.

Sa observam de asemenea fereastră graficelor in care se pot reprezenta grafic diferite marimi ce caracterizeaza modelul simulat. Inainte de inceperea unei noi rulari se merge la: Window, Simulation Window, Plot Window.

De exemplu putem reprezenta grafic lungimea unei cozi. Pentru identificarea acestei lungimi se foloseste SNA-ul Q\$Nume_coadă (in cazul nostru Q\$Coadă), unde \$ este un delimitator. In fereastră deschisa se completeaza “lungimea cozii” la Label si “Q\$Coadă” la Expression. Se tasteaza Plot, apoi Memorize, apoi OK. Apare graficul cu cele 2 axe. Se alege si Memorize pentru a folosi aceasta reprezentare grafica si pentru rulari ulterioare. Se da din nou START 100.

Sa se studieze fereastră expresiilor: Window, Simulation Window, Expression Window. In fereastră deschisa se completeaza “timp” la Label si “c1” (ceasul simularii) la Expression. Se tasteaza View, apoi Memorize. Observam ca la “Memorized Expressions” apare si lungimea cozii, se selecteaza View, apoi OK. Cu fereastră expressiilor deschisa se mai da odata START 100.

Se pot vedea valorile diferitelor expresii si intr-un alt mod: Command, SHOW, c1. In fereastră de compilare, JOURNAL, este afisata valoarea ceasului simularii. Analog, Command, SHOW, QM\$Coadă (lungimea maxima a cozii). Se mai face o rulare cu fereastră expresiilor deschisa.

Exercitii:

1. Sa se modifice programul astfel incat se se formeze o coada de lungime mai mare. Sa se afiseze graficul lungimii cozii si sa se compare cu cel precedent. Sa se observe modificarile valorilor din raport.
2. Sa se modifice programul astfel incat timpii intre sosiri sa aiba o distributie uniforma pe intervalul [50,58]. Sa se observe modificarile valorilor din raport.
3. Intr-un program pot exista mai multe tipuri de tranzactii. Sa se modifice programul astfel incat la benzinarie sa soseasca doua tipuri de clienti: clienti cu masini rosii, care sosesc la fiecare 58 de unitati de timp, si clienti cu masini albe care sosesc la fiecare 50 de unitati de timp. Sa se ruleze programul pentru 100 de masini albe. Cate masini rosii au fost generate? Cate masini rosii au trecut prin sistem?