

Universitatea din București
Facultatea de Matematică și Informatică

CURS nr. 8 – TEHNICI DE SIMULARE

Modele de simulare

Lect. dr. Bianca Mogoș

Conținut

1. Principiile modelării și simulării
2. Generalități despre limbajul GPSS

Noțiuni de bază – Definiție model (1)

- ▶ Oamenii de știință și inginerii încearcă *să înțeleagă, să dezvolte și să optimizeze sisteme.*
- ▶ O *problemă* care apare în analiza unui sistem este *complexitatea* acestuia.
- ▶ Folosirea unor *descrieri simplificate* ale sistemelor – *modelele* – reprezintă o *soluție* care permite abordarea unor sisteme complexe.
- ▶ *Definiție:* Pentru un observator B , un obiect A^* este un *model* al obiectului A dacă B poate folosi A^* pentru a răspunde întrebărilor care îl interesează referitoare la A .
- ▶ *Modelul cel mai bun* este *modelul cel mai simplu* care își îndeplinește obiectivul de a ne ajuta să înțelegem un sistem complex și de a rezolva anumite probleme.

Principiile modelării și simulării

- ▶ Cuvântul *simulare* este de origine latină și înseamnă *capacitatea de a reproduce ceva*.
- ▶ Termenul de simulare a fost introdus de *John von Neumann* la începutul anilor '40.
- ▶ *Modelul de simulare* este o extindere a unui model matematic prin descrierea relațiilor dintre componentele modelului.
- ▶ *Simularea* reprezintă aplicarea unui model (numit model de simulare) pentru a descrie comportamentul și evoluția unui sistem real în timp.

Schema de modelare și simulare

Definiții

- ▶ Definirea problemei pe care ne propunem să o rezolvăm sau a întrebării la care trebuie să răspundem
- ▶ Definirea sistemului – o colecție de obiecte/entități ale căror proprietăți vrem să le studiem

Analiza sistemului

- ▶ Identificarea părților sistemului relevante pentru problemă

Modelarea

- ▶ Dezvoltarea modelului asociat sistemului

Simularea

- ▶ Aplicarea modelului în studiul problemei
- ▶ Dezvoltarea unei strategii pentru rezolvarea problemei

Validarea

- ▶ Verificarea dacă strategia elaborată în pasul de simulare rezolvă problema unui sistem real

Construcția unui model de simulare (1) – Ceasul simulării

- ▶ *Ceasul simulării*: asigură *ordonarea* corectă în timp a *evenimentelor* create de model și uneori ajută la implementarea condiției de terminare a simulării.
- ▶ *Evenimentele* corespund unor modificări în sistem; de exemplu modificarea valorilor unor variabile care se calculează sau se generează prin instrucțiuni ale modelului
- ▶ Ceasul pornește cu valoarea zero la începutul simulării; simularea se realizează până când ceasul atinge o valoare dată inițial T_{max}
- ▶ O *clasificare* a ceasului simulării
 - ▶ *ceas cu creștere constantă*: ceasul crește cu o unitate de timp c constantă, apoi se prelucrează toate evenimentele apărute pe intervalul de timp de lungime c
 - ▶ *ceas cu creștere variabilă*: ceasul crește cu valoarea ce corespunde apariției primului eveniment următor, apoi programul prelucrează evenimentul și se reia ciclul simulării

Construcția unui model de simulare (2) – Agenda simulării

- ▶ *Agenda* conține elementele/variaibilele asociate evenimentelor memorate de modelul de simulare
- ▶ Agenda *A* se compune din două părți:
 - ▶ *AEC – agenda evenimentelor curente* (care au timpul de apariție egal cu valoarea ceasului)
 - ▶ *AEV – agenda evenimentelor viitoare* (care au timpul de apariție ulterior valorii curente a ceasului)
- ▶ Algoritmul de simulare prelucrează numai evenimentele din AEC; prelucrarea unui eveniment înseamnă fie determinarea apariției unui nou eveniment (ce se memorează în AEV) sau modificarea unei stări, fie distrugerea unui eveniment (ștergerea) lui din agendă.

Construcția unui model de simulare (3) – Algoritmul simulării

- ▶ *Algoritmul simulării* actualizează agenda prin *interacțiunea acesteia cu ceasul*
- ▶ Într-un *ciclu al simulării* ceasul este actualizat, se selectează din agenda *A* evenimentele care fac parte din AEC și se prelucrează aceste evenimente până când AEC devine vidă.
- ▶ Atunci ceasul este crescut din nou și se reia *ciclul de simulare*.

Exemplu: Sistemul de așteptare

- ▶ *Sistemul de așteptare* este o parte a lumii reale în care se produc *aglomerări*.
- ▶ *Entitățile componente ale sistemului*:
 - ▶ *clienții* care sosesc în sistem
 - ▶ *stațiile de servire* (care servesc după anumite reguli clienții)
- ▶ Dacă sosesc mulți clienți și stațiile de servire nu pot să îi servească se formează *cozi de așteptare* în sistem.
- ▶ Administratorul sistemului trebuie să construiască sistemul a.î. nici clienții să nu *aștepte* mult până primesc serviciul, dar nici stațiile să nu *lenevească*, deoarece ar putea produce pierderi proprietarului.
- ▶ *Aplicații ale sistemelor de așteptare*: în comerț, în managementul sistemelor de comunicații și de transport, dar și în dirijarea și funcționarea în timp real a rețelelor de calculatoare.

Continuare exemplu: Modelul asociat sistemului de așteptare

Un model de așteptare conține de obicei următoarele elemente:

- ▶ *Variabile de intrare (cunoscute, de obicei aleatoare):*
 - ▶ TI : timp de intersosire al clienților
 - ▶ TS : timp de servire al unui client
- ▶ *Variabile de ieșire (necunoscute):*
 - ▶ TA : timp de așteptare (sau LC : lungimea cozii)
 - ▶ TL : timp de lenevire (sau NSL : numărul stațiilor care lenevesc)
- ▶ *Scopul modelului:* cunoscând repartițiile de probabilitate ale variabilelor AT și ST vrem să obținem informații despre WT și TID și să se stabilească cum trebuie să se realizeze ST a.î. să se optimizeze o anumită funcției criteriu.

Generalități despre limbajul GPSS

- ▶ *Limbajele de simulare* sunt limbaje de programare care implementează elementele de bază ale simulării: *ceasul simulării* și *gestionarea memoriei*.
- ▶ *Utilizatorul* se ocupă de *descrierea evenimentelor* și prelucrarea lor.
- ▶ Exemplu de limbaj de simulare: *GPSS (General Purpose System Simulator)*
- ▶ *Entități ale modelului*:
 - ▶ *Blocuri*: entități care descriu activități
 - ▶ *Tranzacții*: sunt elemente create printr-un bloc GENERATE care parcurg blocurile modelului/sistemului secvențial și în funcție de acțiunea blocurilor accesate.
 - ▶ *Stații de servire* sau *facilități*: resurse care oferă un singur serviciu (adică unei singure tranzacții)
 - ▶ *Multistații de serviciu* sau *depozite*: resurse care oferă servicii mai multor tranzacții simultan
 - ▶ *Entități de calcul*: variabile, funcții
 - ▶ *Entități statistice*: cozi, histogramme

Structura instrucțiunii GPSS

Etichetă	Numele blocului (Cuvânt cheie)	Parametri (separați prin virgulă)
Opțională (pt. referiri)	Verb imperativ indică acțiunea blocului	A,B,C,...

Blocuri de creare și distrugere de tranzacții:

- ▶ *GENERATE* – creează tranzacții. Are forma generală:

GENERATE A,B,C,D,E

- ▶ A – interval mediu între sosiri
- ▶ B – abaterea
- ▶ C – momentul sosirii primei tranzacții
- ▶ D – numărul maxim de tranzacții generate
- ▶ E – prioritate (valoare implicită = 0)

- ▶ *TERMINATE* – determină distrugerea unei tranzacții. Are forma generală:

TERMINATE A

- ▶ A – decrementează parametrul dat la START (în fereastra care apare în urma rulării programului). De exemplu dacă avem "START *nr*" atunci "*nr*" devine " $nr = nr - A$ ". Programul rulează cât timp " $nr > 0$ ".

Blocuri de acțiune

- ▶ *SEIZE/RELEASE* – se folosesc împreună și determină ocuparea/eliberarea unei resurse denumită facilitare; de asemenea creează o facilitare, numele facilității fiind dat ca parametru. Au forma generală:

SEIZE resursă

.....

RELEASE resursă

- ▶ resursă – este numele facilității

- ▶ *ADVANCE* – determină oprirea/blocarea tranzacției care accesează acest bloc un interval de timp dat de parametri. Are forma generală:

ADVANCE A,B

- ▶ A – timp mediu de blocare/întârziere a tranzacției
- ▶ B – abaterea

Blocuri pentru obținerea de statistici

- ▶ *QUEUE/DEPART* – se folosesc împreună și determină obținerea de statistici referitoare la coada de așteptare. Au forma generală:

QUEUE coada

.....

DEPART coada

- ▶ coada – este numele cozii de așteptare

Bibliografie I



I. Văduva (2004), *Modele de simulare: note de curs*, Editura Universității din București, București