

SIMULA 67 SZIMULÁCIÓS ALKALMAZÁSÁRÓL EGY TELEFONFORGALMI PROBLÉMA KAPCSÁN

Knuth Előd

Intézetünk Valószínűségszámítási és Matematikai Statisztikai Osztálya a Beloianisz Híradástechnikai Gyár megrendelése alapján szimulációs programot készített ismétléses telefonhívások által terhelt telefonközpontok vizsgálatára. A szimulációs program SIMULA 67 nyelven készült a CDC 3300-as gépre.

Az alábbiakban röviden ismertetjük a problémát, majd néhány olyan dologra hívjuk fel a figyelmet, mely minden hasonló szimulációs feladatnál felmerül.

A teljes szimulációs programot, továbbá a szimuláció segítségével nyert eredményeket és ezek értékelését ebben a cikkben nem közöljük. Ezt a BHG Telefonfejlesztési Osztálya a közeljövőben publikálni fogja.

Ez a cikk két szempontból tarthat érdeklődésre számot:

1. SIMULA 67 szimulációs subset alkalmazása. (A probléma ugyanis tipikus példa a szimulációs lehetőségek alkalmazására.)

2. Időben lejátszódó parallel folyamatok szimulációja. (Az adott probléma megoldására ugyanis a SIMULA által nyújtott quasiparallel sequencing nagyon hatásos eszköz.)

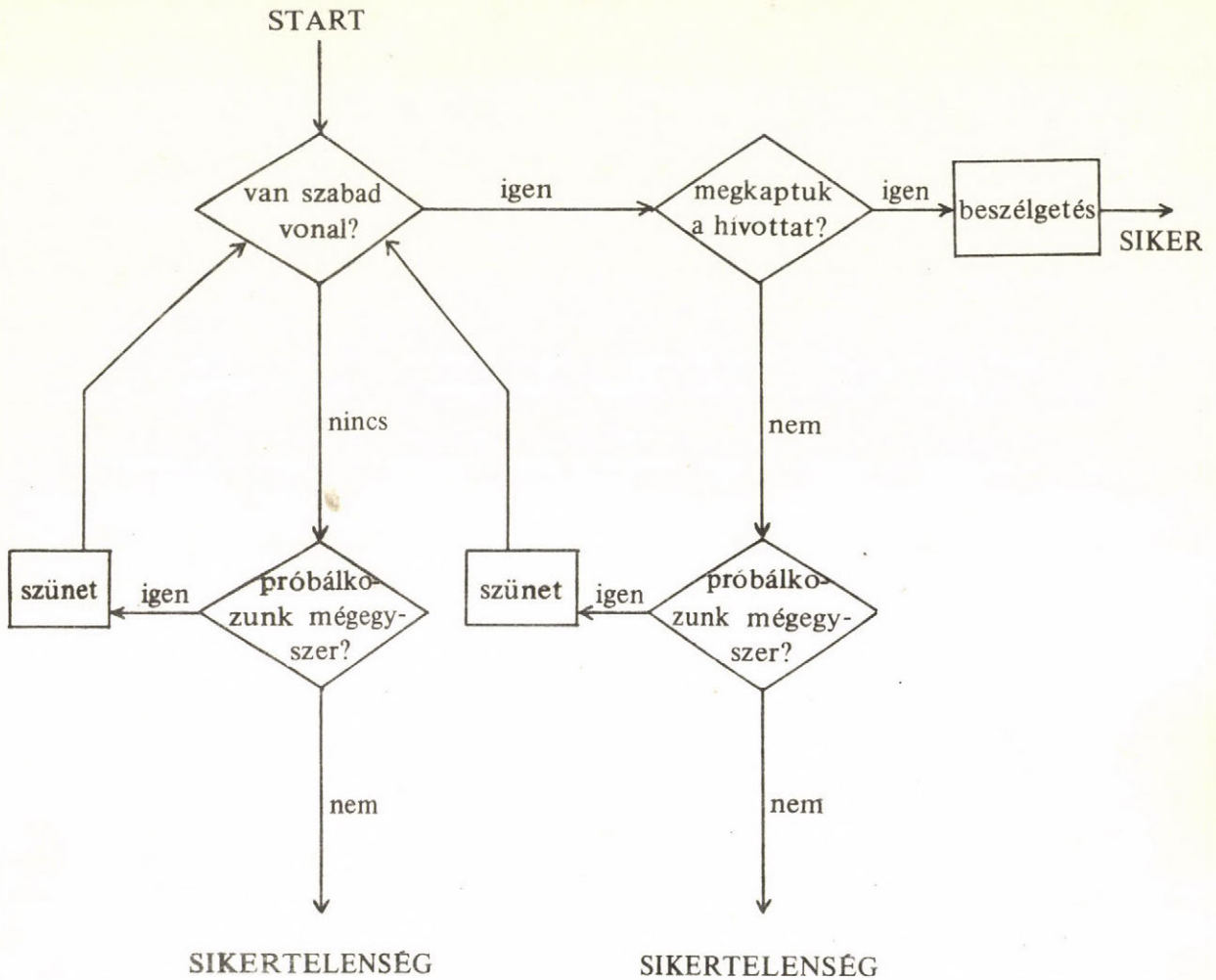
A feladat

Elegendő mindenkinek saját hétköznapjaira gondolni, hogy rögtön észrevegye:

A telefonközpontokat, mint tömegkiszolgáló rendszereket valójában nem egyszerűen hívások, hanem beszélgetési igények terhelik, melyek több hívásból álló sorozatokat is jelenthetnek, ha valamilyen okból az igény azonnali kielégítése akadályokba ütközik. Ilyen ok lehet az, hogy a központ terheltsége miatt a kapcsolás nem tud létrejönni, vagy létrejön, de a hívott mással beszél, esetleg távol van.

Ilyen esetekben a hívó, a sikertelenség okától és eddigi kísérletének számától függően, bizonyos valószínűséggel bizonyos idő múlva megismétli hívását, vagy pedig feladja a további küzdelmet. Ezekre a valószínűségekre és az újrAhívási idő eloszlására vonatkozóan a különféle üzemeltető és fejlesztő cégek ma már nagy mennyiségű mérési adattal rendelkeznek.

A fentiek alapján egy igény működését az alábbi blokk-sémában vázolhatjuk fel:



SIMULA modell

A fentieknek megfelelően az igények szerkezetét, működését az alábbi formában írhatjuk le:

process class hívás;

begin integer ismétlés;

vonal vizsgálat: ismétlés:= ismétlés + 1;

if szabad vonalak száma=0 then

begin if draw (ismétlési valószínűség 1 [ismétlés],.) then

begin hold (szünet);

```
    go to vonal vizsgálat
  end
  else eltávozás
  end
else
if hívott jelentkezik then beszélgetés
else begin if draw (ismétlési valószínűség 2 [ismétlés],) then
    begin hold (szünet);
        go to vonal vizsgálat
    end
end
end hívás;
```

Ha ebben a leírásban a csak sematikusan feltüntetett akciókat már pontosan kidolgoztuk, a szimuláció az alábbi egyszerű programmal bonyolítható le:

```
while time < szimuláció határa do
  begin activate new hívás;
    hold (negexp (beérkezési sűrűség,))
  end;
```

A "hívás" leírására itt megadott process természetesen nem elegendő a kiszolgáló rendszer működésének értékelésére, hiszen hiányoznak belőle azok az utasítások, melyek a vizsgáló karakterisztikákra vonatkozó mintavételeket és számlálásokat végzik. Ezeket értelemszerűen kell a megfelelő helyeken elhelyezni. Az általunk megadott process természetesen ettől függetlenül helyesen szimulálja a vizsgálandó folyamatot, csupán nem ad életjelt magáról.

A megfigyelés problémája

Ebben a pontban egy olyan problémára utalunk, mely minden hasonló feladat megoldásakor fellép, és megmutatjuk, hogy a SIMULA 67 ennek megoldására milyen eszközöket nyújt.

A szimuláció során nyert eredmények általában sztochasztikus folyamatokon értelmezett funkcionálok becslései, melyeknek értékét vagy a teljes realizációk figyelembevételével, vagy statisztikai mintavételek alapján számítjuk ki. A szimulációs folyamat elindításakor a rendszer általában valamilyen különleges, szélsőséges állapotból indul ki, és csak bizonyos idő elteltével válik stacionáriussá.

Nyilvánvaló, hogy a fenti számítások nem lesznek reálisak, ha ezt a kezdeti időszakot is tartalmazzák.

Most bemutatunk néhány programozási fogást, mellyel ez az elkülönítés igen egyszerűen megoldható.

1. Mintavétel, mint "process".

Ha valamilyen karakterisztikát bizonyos időszakonként végrehajtott mintavételek útján akarunk becsülni, elegendő az, ha a mintavételt elvégző utasításokat egy process osztályban írjuk le, ezáltal aktivizálását az időtengely szerint szabadon vezérelhetjük.

```
process class minta;  
  begin array eloszlás [ : ];  
    C: mintavételi utasítások;  
    hold (mintavételi időköz);  
    go to C  
  end;
```

Ebben az esetben a "bemelegítés" különválasztása triviálisan oldható meg generáláskor:

```
activate new minta delay bemelegítés;
```

2. "virtual"

A vizsgálandó karakterisztikák sok esetben olyanok, hogy megfigyelésük csak magában a "hívás" process-ben lehetséges, mert annak működésével kapcsolatos eseményekre vonatkoznak. Ebben az esetben a regisztrálást végző utasítások helyett virtuális eljárásokat alkalmazunk, majd bevezetünk egy speciális "hívás" osztályt, mely semmi másból nem áll, csupán ezen eljárások tényleges definíciójából:

```
process class hívás;  
  virtual: procedure A; . . . stb . . . ;  
  begin  
  .  
  .  
  . (a már ismerttetett törzs)  
  .  
  .  
  end;
```

```
hívás class valódi hívás;  
  begin  
  procedure A; . . . stb.  
  .  
  .  
  . (az eljárások tényleges definíciói)  
  .  
  .  
  end;
```

Szimulációs programunk ezekután így fest majd:

```
while time < bemelegítés do
  begin activate new hívás;
    hold (beérkezési időköz);
  end;
while time < szimuláció határa do
  begin activate new valódi hívás;
    hold (beérkezési időköz);
  end;
```

3. "call"

Az előbbi kérdés egy másik úton is megoldható, mely abban áll, hogy a megfigyelések elvégzése helyett egy objektum hívását iktatjuk be a process-be call segítségével, és a bemelegítés időtartama alatt ezt az objektumot "üresen" tartjuk, majd alkalmas időpontban beletöltjük a szükséges utasításokat.

Ennek vázlata a következő:

```
process class hívás;
  begin
  .
  .
  .   call (regisztrátor);
  .
  .
  end;
ref (regisztráló séma) regisztrátor;
class regisztráló séma;
  begin
    C: detach;
      inner;
      go to C
  end;
```

regisztráló séma class A típusú megfigyelés;

```
begin
.
.   megfigyelést elvégző számítások leírása;
.
end;
```

Ebben az esetben szimulációs programunk a következő alakú:

```
regisztrátor: – new regisztráló séma;
while time < bemelegítés do
```

begin

.

.

end;

regisztrátor: – *new* A típusú megfigyelés;

while time < szimuláció határa *do*

begin

.

.

end;

Irodalom

- [1] Dahl, O. J., Myhrhaug, B. and Nygaard K., SIMULA 67 Common Base Language (Revised edition, Oslo, 1970).
- [2] Knuth, E., "Egy telefonforgalmi probléma vizsgálata Monte Carlo módszerrel" Információ Elektronika (1968).
- [3] Laborczi, Z., SIMULA 67 (Infelör jegyzet, Budapest, 1974).
- [4] Rogeberg, T., Simulation and simulation languages (Oslo, 1973).

Summary

On the application of the simulation subset of the SIMULA 67 language, a tele-traffic problem

The paper shows a typical application of the simulation subset of the SIMULA 67 language, and some special programming technics to obtain correct statistics of the simulated process.

Резюме

О применении языка СИМУЛА 67 связано с одной телефонной системы

В этой работе мы показываем одно типическое применение языка СИМУЛА 67, и методы удорки статистической обсервации моделированных систем.