

NJSZT

MŰSZAKI ÉS TERMÉSZETTUDOMÁNYI EGYESÜLETEK SZÖVETSÉGE

NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG

RENDSZERELMÉLET
KONFERENCIA '79

A RENDSZERELMÉLET ALKALMAZÁSAI

RENDSZEREK SZIMULÁCIÓJA

SOPRON, 1979. szeptember 2-5.



TARTALOMJEGYZÉK

GÁSPÁR ANDRÁS, CSÁKI PÉTER, VISONTAY GYÖRGY

A szimulációs módszer és a SIMULA 67 nyelv 3

ÁGOSTHÁZY MARGIT, GOSZTONY GÉZA, NAGY ROZÁLIA,
RETNE, FÓTI MÁRIAA SIMULA 67 utánzási nyelv alkalmazása telefonrend-
szerek forgalmi vizsgálatához 37

KUN ISTVÁN

Az IGYR-630 integrált gyártórendszer szimulációja . . 54

VÁSÁRHELYI BOLDIZSÁR

A közuti forgalom szimulációs vizsgálatának előnyei . 57

HANGOS KATALIN

Diszkrét idejű sztochasztikus szabályozott rend-
szerek szimulációja a SIMULA nyelv segítségével . . . 64

CZULEK ATTILA

Szakaszos és folyamatos elemeket tartalmazó bonyo-
lult vegyipari rendszerek szimulációja 74

CSÁKI PÉTER

Folytonos-diszkrét kombinált szimulációs rendszer-
vizsgálatok 84

KNUTH ELŐD, SÁRKÓZY ANDRÁS

Diszkrét vezérlési rendszerek modellezése 96

Gáspár A. - Csáki P. - Visontay Gy.:

A szimulációs módszer és a SIMULA 67 nyelv

1. A szimulációs módszer elvi lehetőségeiről és korlátairól

Szimulációval kapcsolatos tudományometriai adatokat közül [1]. E szerint 23 társulat specializálódott szimulációra, 80 bibliográfia foglalkozik különböző területeivel, több mint 1300 doktori disszertáció készült a témáról az Egyesült Államokban és Kanadában. Az utóbbi szám évente 200-zal nő. 300 felett van az angol nyelvű szimulációs könyvek száma, 18 folyóiratot és több mint 100 konferencia kiadványát tanulmányozhatják az érdeklődők e tárgyban. Évente kb. 12 konferenciát rendeznek és több mint 1000 szakkifejezés kialakulása jelzi a téma nagykorúsodását. Számptalan felhasználási területen száz számra készülnek a szimulátorok. A cikk szerint 20 feletti a szimulációs nyelvek száma. Ennek ellenére néhány kutató a szimulációs módszert tudományos/tudománytalan divatnak tartja. Előítéletüket csak erősíti az, hogy mások e terület tudományos létjogosultságát nem magából a módszer lehetőségeiből vezetik le, hanem HELYEJTE a fentiekhez hasonló tudományometriai érveket sorakoztatnak fel, például a szimulációval külföldön foglalkozó kutatók számát.

Meg kell mondanunk, hogy a kételkedés valóban jogos. Ugyanis ezekkel a valóban imponáló számokkal akár a "tudományos" divatot is mérni lehetne. A szimulációs módszer tudományos létjogosultságát tehát sokoldaluan bizonyítani kell.

A rendszermodellezés célja egy rendszer működését, folyamatok interakcióit, időfüggő eseményeit, eseménysorozatait, tömegjelenségeit az eredeti rendszer helyett annak modelljén vizsgálni. Rendszermodellek matematikai vizsgálata rendszerint közönséges és parciális differenciál egyenletek, sztochasztikus folyamatok, sorbanálláselmélet, egészértékű programozás, stb. segítségével történik. Néha a matematikai módszerekkel történő vizsgálat csődöt mond, mert

- a rendszermodell matematikailag kezelhetetlen /egyelőre nincs matematikai modell, egyelőre ismeretlen az egzakt megoldás, nem létezik egzakt megoldás, a megoldást jelentő matematikai algoritmus végrehajtása irreálisan sok műveletet igényel, stb./
- a rendszermodellt a matematikai kezelhetőség érdekében túl-egyszerűsítették /már nem annak a rendszernek a modellje/

Ezekben az esetekben a tudományos rendszermodellezés a teoretikus matematikai vizsgálat helyett /és mellett!/ empiriára, szimulációra kényszerül. A szimulációs módszerek éppen egy-egy teoretikus rendszermodellezési korlát empirikus leküzdése érdekében alakulnak ki /jó! tükrözik ezt a támogatásukra kialakult szimulációs nyelvek, pl. MIMIC, GPSS,.../. A szimulációs módszer tudományos értékű felhasználási területei éppen ott kezdődnek, ahol a matematikai modellezés pillanatnyi határa huzódik, ahol a matematikai

vizsgálat pillanatnyilag csődöt mondott. A szimulációs módszer állandó visszavonulásban van a matematika által ujjonnan meghódított területekről, miközben egyre növekvő számú új rendszermodell szimulációs vizsgálatára kényszerülünk.

A szimulációs módszer lehetőségei és korlátai jól megvilágíthatók összetevőinek lehetőségein és korlátaikon keresztül. A szimulációs módszer összetevői a szimulációs műveletek. Elemi szimulációs műveleteknek tekinthetők: szimulációs regisztrálás, szimulációs beavatkozás; szimulációs demonstráció /szimulációs bizonyítás/. Összetett szimulációs műveletek, a szimulációs megfigyelés-mérés, a szimulációs kísérlet, a szimulációs hipotézisvizsgálat, a szimulációs predikció /szimulációs előrejelzés/, a szimulációs döntéselemzés. A szimulációs módszer tipikus alkalmazási területei a szimulációs megismerés, a szimulációs tervezés, a szimulációs irányítás-szabályozás. A fenti osztályozás tökéletesítése valószínűleg kulcsot adna a szimulációelmélet kezébe a meglévő szimulációs eszközök kiértékeléséhez és hiányosságaik kimutatásához.

A szimulációs módszer segítségével nemcsak a matematikai vizsgálat, hanem többnyire a rendszer közvetlen vizsgálatának legjellegzetesebb korlátait is leküzdhetjük. Állításunk alátámasztására kiemeljük egy részletesebb elemzés két bekezdését:

A szimulációs mérés/mintavétel/ elvi lehetőségei:

Nagyszámú paraméter mérhetősége. Nagyszámn mérési /mintavételi/ pont lehetősége. Az eredeti rendszerben mérhetetlen paraméterek és pontok mérhetősége.

Beavatkozásmentes mérés /mintavétel/ lehetősége. Korlátlan mérési pontosság lehetősége. Egyidejű mérés /mintavétel/ lehetősége. A rendszert megsemmisítő-átalakító folyamatok sokszori mérhetősége. A mérési eredmények automatikus regisztrálhatósága, szelektálhatósága. Automatikus paramétertípus váltás, mérési stratégia, ill. mérési pont váltás lehetősége. Tul gyors, tul lassu folyamatok mérhetősége...

A szimulációs kísérlet elvi lehetőségei

Korlátlan mérvű beavatkozás lehetősége. A szimulációs idő reverzibilitása. Pontosan ugyanazon állapotban történő korlátlan számú alternatív beavatkozás következményeinek megfigyelhetősége, mérhetősége. Elvileg lehetetlen, tiltott kísérletek /pl. orvos etikailag/ ill. beavatkozások végrehajthatósága. Korlátlan számú egyidejű beavatkozás lehetősége. A kísérleti eredmények automatikus regisztrálhatósága, szelektálhatósága. Korlátlan mérvű adaptivitás lehetősége a megfigyelési, mérési, kísérleti eredményekhez. Kísérlet lehetősége a valóságban megismételhetetlen /rendszert átalakító, megsemmisítő/ folyamatok korlátlan számú reprodukciójával. Kísérletezés lehetősége tul gyors, tul lassu folyamatokkal. Tetszőleges pontosságú kísérlettervezés lehetősége...

2. Az eseményközpontú szimulációs nyelvek rendszer- elemzési képességének kiértékeléséről.

Az eseményközpontú szimuláció a rendszermodell tulajdonságaival jellemezhető a legjobban: Minden állapotváltozás diszkrét időpontokban történik. Kulcsfogalom az esemény, mely egyidejű egymást követő állapotváltozások sorozata. A modellidő eseményről eseményre ugrik, monoton nő.

Az eseményközpontú szimuláció céljára leggyakrabban használt 4 "általános" szimulációs nyelv a GPSS, a SIMULA, a GASP és a SIMSCRIPT. 1976-ban nagyszámú szimulációs nyelv jó és rossz tulajdonságainak összehasonlítása útján egy kiterjedt kritériumrendszert állított össze W.Kreutzer [2] és segítségével kiértékelte a fenti 4 nyelvet. 8 különböző súlyú kritériumcsoportot alakított ki:

- nyelvtanulás könnyedsége
- rendszerleíró képesség
- a programozási rendszer lehetőségei
- a fordítóprogram jellemzői
- a futtató rendszer /kísérőprogram/ jellemzői
- biztonsági jellemzők + belvási könnyítések
- modellérvényesítési szolgáltatások
- szolgáltatások a modellel történő kísérletezésre

Kreutzer értékelése világosan rámutat a SIMULA erejére, bár a kritériumrendszer - legalábbis a SIMULA rovasára - erősen torzít. A mintegy 200 címszót tartalmazó, empirikusan összeállított kritériumrendszer

teoretikus felépítése és kiegészítése - úgy hisszük - ma is a szimulációelmélet egyik aktuális problémája.

Egy későbbi teoretikus felépítésnek a rendszerelemzés-modellezés folyamatából kell kiindulnia, végigkövetve a rendszerelemző útját /eredeti rendszermodell-szimulációs modell-számítási modell azaz szimulátor/, levezetve a szimulációs rendszerelemzés támasztotta követelményeket, és számbavéve a szimulációs nyelvek szolgáltatotta lehetőségeket. Példaként egy ilyen teoretikus felépítés két részletét vázoljuk.

A rendszerleíróképeség problémája:

Minden szimulációs nyelvnek minimalizálni kell a "fogalmi távolságot" a vizsgált rendszer és a rendszermodell "világképe" között. Ezért a rendszerelemző segítése céljából minden nyelvnek egy többé-kevésbé általános rendszerszemléletet kell kialakítania és ennek megfelelően definiálni a nyelv legfontosabb konstrukcióit.... Milyen rendszerszemlélet jelenik meg a különféle nyelvek utasításkészletében? Milyenek az ütemezési lehetőségek?...

A szimulációs szolgáltatások problémája:

A rendszermodell működésének indítási fázisa bonyolult, gépidőt rabló folyamat. Ráadásul az indítási fázis gyakran kívül esik a rendszervizsgálat körén... Mikor és hogyan kapcsolható be és ki a megfigyelés-mérés-kísérlet? /indítási-megállási szabályok programozhatósága/... Hogyan jeleníthető meg a megfigyelés-mérés-kísérlet eredménye? Hogyan realizálható a beavatkozás? /input-output-statisztika-interaktív kapcsolat lehetőségei/

3. A SIMULA 67 rendszerszemléletéről és modellkezeléséről

Gyakori, hogy a számítástudományi kutatók, programozók, vagy rendszerelemzők SIMULA 67-tel kapcsolatos ítéleteiket anélkül fogalmazzák meg, hogy megértenék, megismernék a SIMULA 67 rendszerszemléletét és modellkezelését. E téves ítéletek gyűjteménye bizonyára a tudománytörténet humoros fejezeteit gazdagítja majd. Példaként ehelyütt csak egy "ítéletet" említünk.

A SIMULA 67 az ALGOL nyelvcsalád tagja, tartalmazza az egész ALGOL 60 nyelvet. Emiatt felületes "ismerői" rögtön kész "ítéletet" alkotnak róla: Az ALGOL 60 kiterjesztése. Valóban az, amiként a kutya is a fark kiterjesztése. Sőt ugyanolyan nézőpontból bátran állitható az is, hogy a fark csóválja a kutyát.

A SIMULA nyelv értékeit /rendszerszemléletét, modellkezelését, kifejezőerejét, hatékonyságát stb./ 1967 óta még egyetlen eseményközpontú szimulációra szolgáló nyelv sem haladta meg. Ennek egyszerű magyarázata abban rejlik, hogy a SIMULA 67 tervezése során /1962-68/ a konstruktőrök a kor valamennyi szimulációs nyelvét és programnyelvét kritikai vizsgálat alá vették. Végülis 1967-ben a Norwegian Computing Center az általa szervezett "Common Base Conference" elé tárta az eseményközpontú szimuláció közös alapszabályaira tett javaslatát. A jelentős nemzetközi résztvevő gárda tulajdonképpen az eseményközpontú szimulációs nyelvek konstruktöreiből, implementálóiból és

fő felhasználóiból tevődött össze. A konferencia javaslatára komoly változtatások is születtek: Egy évi kemény munka után 1968-ban elkészült a SIMULA szövegkezelése és input-output kezelése. A nyelvfejlesztés történetét [14], végeredményét pedig a "SIMULA 67 Common Base Language" [3] című jelentés ismerteti.

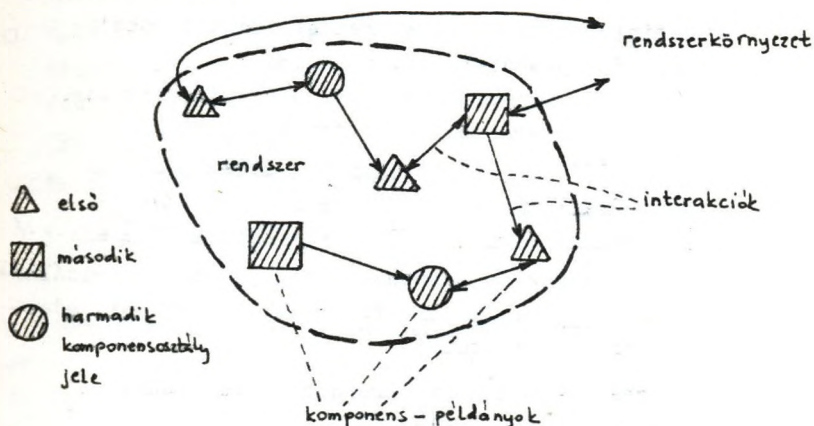
A SIMULA 67 rendszerszemlélete az általa szolgáltatott nyelvi eszközökben tükröződik. Legjobban egy-egy rendszerfogalom és a megfelelő nyelvi eszköz szembeállításával, a "fogalmi távolság" mértékével érzékeltethető. Tekintsük át tehát először a rendszerek legfontosabb tulajdonságait, a legfontosabb rendszerfogalmakat.

A rendszerek alkotóelemei a komponensek. A komponensek működnek, változnak, bennük folyamatok zajlanak. Bizonyos komponenspárok között meghatározott időben és/vagy meghatározott feltételek teljesülése esetén interakciók /hatások, kölcsönhatások/ lépnek fel.

A komponensek legtöbbször maguk is rendszerek. A rendszer jól jellemezhető komponenseivel, komponensszerkezetével és a rendszer környezetével, továbbá mindezek működésével, változásával, illetve interakcióikkal.

A rendszer komponensei - közös tulajdonságaik alapján - komponensosztályokat alkotnak. A rendszer komponensszerkezetében általában ismétlődések vannak,

vagyis ugyanazon komponensosztályba általában több komponens-"példány" tartozik. Tekintsük például a következő három komponensosztályt és hét komponens-példányt tartalmazó rendszert:



Számos rendszer "működése" alatt új komponens-"példányok" jönnek létre /lépnek a rendszerbe/ illetve szűnnek meg /távoznak a rendszerből/.
Vagyis számos rendszer működése során változik a rendszerhez tartozó komponens-"példányok" száma.

Vegyük sorra a SIMULA 67 kiváló rendszer leiró képességének legfontosabb bizonyítékait.

- Rendszerek és komponensosztályok definiálhatók /class deklaráció/
- Definiálhatók a rendszerek és komponensosztályok állapotátározói, funkciói /class deklaráció törzsében szereplő adatjellegű és eljárásjellegű attributumok, Lehetséges típusaik: integer, real, Boolean, character, text, ref(), integer array,..., procedure, integer procedure, real procedure,.../
- A rendszerek és komponensosztályok parametrizálhatók /class deklaráció formális paraméterei, adatjellegű attributumai/
- Leírható a rendszer-alrendszer hierarchia /lokális class deklaráció/
- Leírhatók a rendszeren illetve komponenseken belüli folyamatok /class deklarációban definiálható eljárásgyűjtemény és forogatókönyv/
- Új komponenspéldányokat lehet létrehozni /new utasítás/
- Új rendszerpéldányokat lehet létrehozni /new utasítás, belépés prefixes blokkba/
- Meg lehet adni az új komponenspéldány, illetve rendszerpéldány születési paramétereit /new utasítás aktuális paraméterei, prefixes blokk aktuális paraméterei/

- Komponenspéldányokat és rendszerpéldányokat el lehet nevezni, lehet rájuk kívülről hivatkozni /ref típusú változók/
- Lehet rájuk belülről is hivatkozni /ref típusú változók, this utasítás/
- Leírhatók a "parallel" működő komponenspéldányok, illetve rendszerpéldányok közötti interakciók /távoli elérés, detach, call, resume, activate, reactivate,.../
- Megszüntethetők a komponenspéldányok, rendszerpéldányok /:- none utasítás, kilépés prefixes blokkból + szemétyűjtés/

A SIMULA 67 rendszerelemzésbeli értékét nem csupán rendszer szemlélete determinálja, hanem modellkezelése is.

Modellkezelése nemcsak a szimulációs nyelvek körében páratlan, de e vonatkozásban is messze a legfejlettebb univerzális programnyelvek /ALGOL 68, MODULA, PASCAL, stb./ előtt jár.

Tekintsük át először a rendszermodellezés-elemzés legfontosabb jellemzőit.

A rendszermodellező a logikából ismert hierarchikus fogalomrendszerhez hasonló, hierarchikus modellrendszer [15] alakít ki a "bottom-up" általánosítást, illetve a "top-down" specializálás útján. Például:

- A nyelv jól kiválasztott standard modell-hierarchiákat tartalmaz

/class BASICIO : standard input/output modell

class SIMSET : standard listakezelési-sorkezelési modell

SIMSET class SIMULATION: standard folyamatkezelési modell/

4. Szimulációs vizsgálatok SIMULA 67 nyelven

Az alábbiakban az 1973 óta negyedévenként megjelenő SIMULA Newsletter [4] című folyóirat alapján címszavakkal utalunk a legfontosabb szimulációs eredményekre, irányokra.

A leggyakrabban szimulált rendszerek

Számítástechnikai rendszerek, adatbázis kezelő rendszer, operációs rendszerek, real-time rendszerek, terminál hálózatok, számítógép hálózatok kikötők, vasuti rendezőpályaudvarok, közuti forgalom
 gyárak/acélgyárak, erőművek/, olajfuróállomás
 automaták, számítógépek, telefonközpontok
 termelési folyamat/gyártás, karbantartás, raktári IO/
 áruházak, kereskedelmi vállalatok
 számítóközpontok, szervezetek/posta/
 konfliktusok /piaci, katonai, fejlődési/
 biológiai rendszerek /idegrendszer/
 kezelhetetlen matematikai modellek /sorbanállás,
 szinkronizálás, alternatívákat tartalmazó PERT/CPM/

A legfontosabb rendszerelemzési célok

Kapacitásbecslés, megbízhatóságanalízis, szűk keresztmetszetek meghatározása, költségbecslés /pl. beruházási, fenntartási/, igény-előrejelzés / pl.fo-

gyasztási/, ütemezés, működési stratégia kiértékelés-meghatározás, szervezetkiértékelés.

A SIMULA Newsletter szimulációs rendszervizsgálatokról szóló közleményei

A hivatkozások alakja: (intézmény évfolyam/szám.oldal)

Adatbáziskezelő rendszerek

- adatbáziskezelő szimuláció (Univ. of Calgary 1977/2.10; Royal Ins. of Technology, Stockholm 1977/1.5)

Operációs rendszerek

- tervezés SIMULÁ-ban (DEC 1975/4.21; OLIVETTI 1977/3.20)
- szimuláció és "hangolás" SIMULÁ-ban (Univ. Pierre et Marie Curie 1975/3.7; Univ. of Calgary 1975/3.8; 1977/2.10; DEC 1975/3.8; Univ. of Bonn 1976/3.4; NCC 1976/3.4; Univ. Libre de Bruxelles 1977/3.7)

Real-time rendszerek

- tervezés, szimuláció, "hangolás" SIMULÁ-ban (Swedish Nat. Def. Res. Ins. 1977/3.12)

Terminálhálózatok, számítógéphálózatok

- tervezés, szimuláció (Karlsruhe Nuclear Res. Center 1974/2.7; 1974/3.8; Univ. of Calgary 1976/3.13; Royal Ins. of Technology, Stockholm 1977/1.5; Univ. of Oslo 1978/3.13)

Kikötők

- PORTOS kikötő szimulátor: hajóérkezés, kiszolgálás, időjárás, hullámszél, ár-apály, révkalauzok, vontatók, műszakok, biztonsági szabályok, kikötői műveletek,
cél: max. fogadó és tárolókapacitás (Amersfoort, Hollandia 1975/3.11)
- Newcastle kikötői (Maritime Service Board, New South Wales 1977/4.7)
- Jégtörő flotta műveletei: 2 jégtörő 3 svéd tengeri kikötőre,
kérdések: működési rend? nagyobb forgalom jégtörő igénye?
(Swedish Nat.Def.Res.Ins. 1976/3.10; 1976/4.19)

Közúti közlekedés

- input: uthálózat+forgalom leírás. Cél: forgalmi szituációk jellemzése (Univ. of Stuttgart 1974/3.11)

Acélgvár

- Berendezések karbantartásának ütemezése /kenés, ellenőrzés, csere/.
Munkalapok kiállítása a karbantartók számára (ESTEL Company 1974/1.8)
- Gyári vasúthálózat vizsgálata. Cél: szállítási szabályok és irányok, szűk keresztmetszetek meghatározása; területfelszabadítás (ESTEL Company 1974/1.8)

Olajkutatás

- Geológiai szimulációk (Univ. Pierre et Marie Curie 1976/3.9)
- Olajfuróállomás szimulációja (Univ. Pierre et Marie Curie 1978/3.22)

Hirközlés

- Kisvárosi telefonrendszer paraméterbecslése (Univ. of Bologna 1977/3.22)

Termelési folyamat

- Folyékony szappangyártás-kapacitásanalízis tartályok, autoklávok, szivattyuk (COLGATE-PAIMOLIVE 1974/4.14)
- Acélgyártás (INORGA Ins.-Charles Univ. 1977/3.22)

Kereskedelem

- SIMWAP áruháztervezést segítő szimulátor
Néhány fogalom: belső-raktárak, árucikkek, belső-megrendelések, belső-szállítások, belső-raktári készletek
Részfeladat: belső-szállítás tervezés, munkalapírás a szállítómunkások számára (NCC 1973/1.6)

Biológiai rendszerek (Univ. of Freiburg 1976/3.7)

- Idegrendszer (Ins. for Experimental Psychology, Groningen 1976/2.5)

Kezelhetetlen matematikai modellek

- Erőforráskezelés és folyamatszinkronizáció

(Robin Hills
Consultant LTD - Univ. of Bradford 1976/1.4),
kiterjesztett Petri-háló (Karlsruhe Nuclear
Res. Center 1976/3.13)

- Alternatívákat tartalmazó PERT/CPM (SHAPE
Technical Center 1974/3.15)

Eddig öt szimulációs vonatkozású SIMULA-Workshop
volt:

- 1., Continuous and Discrete Simulation in SIMULA
- 2., Design and Tuning of Operating Systems
- 3., SIMULA and Real-Time
- 4., SIMULA and Data Bases
- 5., SIMULA and Design of Simulation Experiments

Ennyi külföldi eredmény láttán joggal merül fel a
kérdés: hol tartunk ma Magyarországon. Ha fel akar-
juk sorolni a legfontosabb eredményeket sajnos köny-
nyű dolgunk van, mivel ma - 1979 áprilisában - csak
egyetlen gépen létezik SIMULA fordítóprogram az or-
szágban. SIMULÁ-ban implementált legfontosabb hazai
szimulátorok témái és céljai a következők:

Hardware tervezés

- parallel processzorok szimulációs vizsgálata (BME)

Távközlési rendszerek

- telefonközpontok tervezése (MTA SZTAKI, BHG)

Operációs rendszerek

- a CDC 3300/MASTER operációs rendszer tárgazdálkodásának vizsgálata installációs paraméterek "hangolása" céljából (MTA SZTAKI)

Számítógéphálózatok

- az MTA tervezett csomagkapcsolt számítógéphálózatának tervezése (MTA SZTAKI) [5] [6] [7]

Közuti közlekedés

- keresztezésmentes útszakaszok forgalmi áramlatainak vizsgálata (VÁTI)

Termelési folyamatok

- integrált gyártó rendszer tervezése (MTA SZTAKI Csepeli Szerszámgépgyár)

Vegyipari rendszerek

- diszkrét idejű sztochasztikus szabályozott rendszerek (MTA SZTAKI)
- összetett rendszerek működésének leírása és szimulációja Petri háló segítségével (MTA SZTAKI)
- szakaszos és folyamatos vegyipari üzemek szimulációs vizsgálata (MTA SZTAKI, BME)
- Szabolcs-Szatmár megye tankautós benzín és gázolaj ellátásának megtervezése (OLAJTERV)

Mezőgazdasági termelési rendszerek

- vetés/betakarítás szimulációja (Keszthelyi Agrártudományi Egyetem)

Biológiai rendszerek

- neuronhálózatok vizsgálata [8], [9], [10]
- nyomhipotézis és adaptivitás vizsgálat
(MTA SzTAKI)
- a Balaton ökoszisztémájának vizsgálata kombinált /folytonos+diszkrét/ szimulációval
(MTA Biológiai Kutató Intézet-MTA SZTAKI-VITUKI)

5. Szimulációs célú kutatási-fejlesztési irányokról

5.1 Szimulációs beágyazásokról

A SIMULA 67 kiterjeszhető nyelv, ami azt jelenti, hogy modellkezelése egyszerű, hatékony és megbízható eszközöket szolgáltat felhasználói modellek definíciójára, könyvtárbahelyezésére, és az előre elkészített könyvtári modellek igen egyszerű használatára. A SIMULA nyelvbe a kiterjesztés eszközeivel célorientált, de mégis az aránylag általános szimulációs rendszereket lehet beágyazni.

Portábilis szimulációs beágyazásokkal kapcsolatos külföldi eredmények a SIMULA Newsletter közlemények alapján:

- GPSS Általános tömegkiszolgáló rendszer szimulációs osztály. Mérete 700 forrásprogram-sor. (Univ. of Montreal 1975/3.3; Univ. of Trondheim 1977/2.4)
- SIMON-75 Felhasználó közeli szimulációs eszközöket ad tömegkiszolgáló-rendszerek szimulációjára (Robin Hills Consultant LTD.-Univ. of Bradford 1975/3.11; 1976/4.6; 1977/2.4)
- DEMOS A SIMON-75 továbbfejlesztése (Univ. of Bradford 1979/1.4.)
- DYNAMO Ipari dinamika szimulációs rendszer (Univ. of Erlangen-Nürnberg 1977/3.15; Technical Univ. of Berlin 1978/3.8)
- CADSIM Folytonos+diszkrét=mixelt szimulációs osztály (Imperial College of Science and Technology 1976/2.3; 1977/2.4)

- COMBINEDSIMULATION A CADSIM továbbfejlesztése
(Univ. of Roskilde 1978/3.8)
- CADSIMCLASS GPSS A CADSIM és a GPSS keresztezése.
Polytonos kapacitások, erőforrások, péld-
ául vegyipar szimulációra. (Univ. of
Trondheim 1977/2.4)
- OASIS Operációs rendszer-adatbáziskezelőrend-
szer szimulátor (Univ. of Calgary
1977/2.10)
- SIMULATIONTRACE (Univ. of Groningen 1978/1.7)

Hazai eredmények:

- DISCOS A CADSIM továbbfejlesztése /MTA SZTAKI/
- CSM A Control System Modelling különféle
vezérlési rendszerek modellezésére,
szimulálására kidolgozott rendszer
/MTA SzTAKI/

5.2 A DELTA rendszerleírónyelvről és szimulátorok dokumentálásáról

A DELTA szigorú szintaxisú, számítógépfüggetlen rendszerleíró nyelv.

(NCC 1974/3.12; 1975/1.11; 1975/3.14; Univ. of Aarhus 1976/3.16). A DELTA rendszerleírónyelv világos bizonyítékot ad arra, hogy nem a matematikai nyelv az egyetlen, mely tudományközi érintkezést tesz lehetővé. Definícióját 1975-ben bocsátotta ki a Norwegian Computing Center [11]. A definíció hasonlít a programozási nyelvekéhez, a nyelv szerkezete és fogalomkészlete pedig a SIMULA 67 általánosítása és bővítése. Pontossága és kifejező ereje nagy segítség lehet a rendszerek leírása, tervezése; tanulmányozása és a róluk szóló kommunikáció során.

Példaként csak a következőt említjük. Néhány hazai, SIMULA nyelvű 1500-2000 soros szimulátor /operációs rendszer, számítógéphálózat, telefonközpont/ dokumentálása - a SIMULA nyelv kiváló, az öndokumentálást megközelítő tulajdonsága ellenére is - jelentős külön munkát jelent. Ugyanis nem elegendő a program egyszerű csatolása a szimulációs eredményhez, mivel az eredményeket ellenőrző tanulmányozó rendszerelemző elveszne a programmegvalósítás részleteiben, hiszen a korlátos számítógép erőforrások és a nyelv ésszerű utasításkészlet korlátai miatt a program írói programozási trükkökre kényszerültek.

A DELTA itt sokat segíthet, mivel nem programozási-, hanem számítógépfüggetlen rendszerleíró nyelv és ezért megszabadítja a rendszerleírást a számítógépes erőforráskorlátok torzító hatásától.

Nem kötik a számítógép memóriakorlátai, ezért korlátlan méretű rendszer leírására alkalmas. Nem kell implementálni, ezért utasításkészletét a rendszerelmélet és nem az implementálás nehézségei, a memória - és sebességekorlátok determinálják. Nincs processor, mely eseményközpontu rendszerleírást és kvázi párhuzamos folyamatvégrehajtást kényszerít ki, tehát a rendszerleírás az eseményeken túl, valóban párhuzamos, folytonos folyamatok definícióját is tartalmazhatja. Nincs művelet végrehajtási sebesség, ezért a leírás korlátlanul számításigényes részeket is tartalmazhat. Nincs gépköltség, lényegtelen a számítás hatékonysága, tehát pl. a legegyszerűbb, legerősebb matematikai definíciót választhatjuk. Nincs szóhossz, tehát a REAL típusu változó értéke nem egy kettédes tört, hanem egy igazi valós szám. Nincs gond a realizálás algoritmusával, így majdnem matematikai egzaktsággal és egyszerűséggel adhatjuk meg a folyamatok indításának és leállításának feltételeit. Viszont sajnos nincs gépi futtatás sem, tehát a rendszer állapotait nem géppel nyomtatott pillanatképek jellemzik, hanem képzeletben rajzolt pillanatképek, melyek az időponttal indexeltek, tehát számosságuk a valósszámok számossága.

A rendszer működését nem kötelező zéró időtartam alatt bekövetkező eseményekkel leírni, hanem leírhatók időben zajló folyamatokkal is. Egy komponens hatása a másokra nem csupán attributum olvasás-módosítás, ill. eseménykiváltás lehet, hanem lehet a másik komponensen zajló folyamat megszakítása is. A DELTA rendszerleírónyelv igen értékes fogalomkészletet szolgáltat a folyamatmegszakítások leírására /megszakítás küldő, megszakítás fogadó, megszakítás behatolás, ellenállás a megszakítással szemben, megszakítási szint, prioritás, megszakítás bevezetése, megszakítás befejezése, megszakítási verem/. E fogalomkészlet rendszerleírásbeli jelentőségét, erejét ma még nem látjuk át. Elképzelhető, hogy rendszerleírásbeli jelentősége összemérhető a megszakítási rendszer számítástechnikabeli megjelenésével.

Rendszerleírással és dokumentálással kapcsolatban ki kell emelnünk egy hazai kutatási eredményt (MTA SzTAKI, 1978) [12]. Mivel a SIMULA 67 célkitűzése volt, hogy rendszerleíró és modellkezelő nyelv legyen, ezért a SIMULA nyelven írt szimulátorokban jól elkülöníthetők a különböző rendszerkomponenseket, illetve modelleket reprezentáló programrészletek. Emiatt a nyelv szabályai alapján könnyen felismerhetők a komponensek modelljei, a komponensek attributumai, a komponensek közötti interakciók, stb. Mindez lehetővé tette egy olyan program elkészítését amely egy szimulátor forrásprogramjából mint inputból, outputként a rendszerdokumentáció egy részletét

állítja elő.

Kézenfekvő a kérdés: Nem lehetne-e hasonló programot készíteni DELTA nyelvű rendszerleírások számítógépes elemzése céljára is? Természetesen igen, mivel a DELTA-nak szigorú szintaxisa van. Így tehát megfelelő programok birtokában a DELTA rendszerleírásból mint inputból számítógéppel egy sereg olyan outputot állíthatnánk elő, amely a rendszertervezők - elemzők munkáját nagymértékben segíthetné. Néhány példa a generálható outputra:

- számítógéppel generált szimulátor
- számítógéppel generált grafikus dokumentumok /blokkdiagramok, folyamatábrák, kapcsolatdiagramok/
- számítógéppel generált szöveges dokumentumok /rendszerkomponensekről, interakcióikról, stb./
- a rendszerterv számítógéppel feltárt logikai ellentmondásai /pl: ütemezési, költség, geometriai, teljesítmény ellentmondások/

5.3 A szimulátorok szerkezetéről, és a nagyrésztességű eseményközpontú szimuláció software eszközeiről

Két egymástól merőben különböző eseményközpontú szimulációs tevékenység software igényeit kell feltárnunk:

- az események, eseménysorok vizsgálatára alkalmas nagyrésztességű szimulációt,
- a tömegjelenségek vizsgálatára alkalmas kisrésztességű szimulációt.

A nagyrésztességű szimuláció tipikus felhasználási területei: Rendszerleírások logikai helyességének vizsgálata. Kritikus események, eseménysorozatok analízise. Algoritmikus működési szabályok célszerűségének demonstrálása.

Kisrésztességű szimulációval vizsgálják tömegkihasználórendszerek esetén például a következőket: teljesítmény, erőforráskihasználtság, átlagos várakozási idők, stb.

A számítástudományban közismert fogalom és jelenség a "kombinatorikai robbanás". Ez a fogalom arra a jelenségre utal, hogy egyes algoritmusok műveletigénye kombinatorikai függvény szerint nő a probléma "méretének" növekedésével. Hasonló értelemben beszélhetünk szimulációs robbanásról is. Mindezek súlyos követelményeket irnak elő a szimuláció software

eszközei /szimulációs nyelv, operációs rendszerek és kiterjesztések, adatbáziskezelők, monitorok/ számára.

Például előzetes adatellenőrzéssel meg kell akadályozni azt, hogy egy hosszabb szimulációs futás adathibával termináljon. A szimulációs erőforrásigények /CPU-idő, memóriaméret, diskterület, nyomtatandó sorok, stb./ előzetes automatikus becslésével meg kell előzni azok az abnormális terminálásokat, melyek valamelyik erőforráskorlát túllépése miatt következnenek be. Mindezek, súlyosbitva a rendszerint nyomasztó memóriahiánnyal, legalább 3 fázisu szimulátorokat /előkészítő-, szimulációs- és kiértékelő fázis/ kényszeritenek ki.

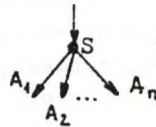
Például az MTA csomagkapcsolt számítógéphálózatának szimulációjához öt fázisú szimulátort készítettünk. Fázisai:

- Konfigurációleírás. Irányítási táblák kezdőértékeinek generálása.
- Rendszerparaméterbeállítás. Élettartamgenerálás. A szimuláció erőforrásigényeinek becslése, előrejelzése.
- Megfigyelés definíció. Riporter programozás.
- Szimuláció a kijelölt időintervallumban.
- Kritikus események és teljesítményjellemzők, vagy megfigyelt folyamatok listázása.

A nagyrészletességű szimuláció megsokszorozza az erőforrásigényeket és ezért mégjobban kielejezi a problémákat. A nagyrészletességű szimuláció támogatására a CDC 3300 SIMULA/MASTER-hez illesztve megterveztük és implementáltuk a Task - és Run Time Library kezelést [13]. Segítségével megoldódott egy sereg korábbi probléma és új rendszerelemzési lehetőségek birtokába jutottunk:

- Csökkenthetők az indítási veszteségek. Megvalósítható pl. az S stacionér állapot bekövetkezése után az A_1, A_2, \dots, A_n alternatívák "olcsó" vizsgálata.

/Ugyanis az S szimulációs állapot kimenthető egy könyvtárba és onnan visszaállítható/



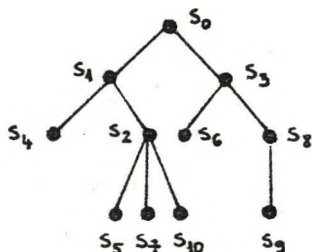
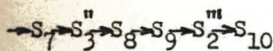
- Szabályozható a szimuláció részletessége. Programozható a megfigyelés, a mérés, a kísérlet. A "fókuszállítás" olcsó visszalépéssel és a megfigyelés-mérés-kísérlet olcsó újraprogramozásával valósítható meg. Pl.:

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_2' \rightarrow S_4 \rightarrow S_5$



- Szimulációs vizsgálatokra is alkalmazható az operációutatóból ismert értékelés és szétválasztás /branch and bound/ módszere.

Például: $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0' \rightarrow S_3 \rightarrow S_1' \rightarrow S_4 \rightarrow S_2' \rightarrow S_5 \rightarrow S_3' \rightarrow S_6 \rightarrow S_2'' \rightarrow$



- Intelligens szimulációs rendszerek írhatók, mivel lecsökkentek a központi memória hiányából adódó problémák, állapotok menthetők és visszaállíthatók, továbbá taskrendszerek szervezhetők.

Hasonló operációs rendszer szolgáltatást több gépen is illesztettek a SIMULA 67 nyelvhez (1975/4.8; 1977/1.16; Norwegian Computing Center 1978/2.10; MTA SzTAKI 1978/2.3)

Az adatbázisok a szimuláció során legalább 3 szerepkörben jelennek meg: input-adatbázis, output-adatbázis és átmeneti adatbázis. Az első esetben a szimulátor egy adatbázisból veszi elő a működéséhez szükséges adatokat, a második esetben a szimulátor a működése során adatokkal tölt fel egy adatbázist. Átmeneti adatbázis jön létre például hierarchikus szimuláció esetén, amikor egy-egy szint szimulációja közben töltődő/feltöltött adatbázist a másik szint szimulációja dolgozza fel.

SIMULA 67 nyelven több univerzális adatbáziskezelő rendszert implementáltak:

- CODASYL típusút - SIMDBM (Swedish Nat.Def.Res.Ins. 1975/3.9)

- relációst - ASTRA (Univ.of Trondheim
1977/3.23)
- hipergráf alapút (Univ. Perre et Marie Curie
1976/3.14)

IRODALOMJEGYZÉK

1. T.I.Ören: Simulation - as it has been, is, and should be. SIMULATION, 1977. pp.182-183.
2. W. Kreutzer: Comparison and Evaluation of Discrete Event Simulation Programming Languages for Management Decision Making
North-Holland Publishing Company, 1976
Simulation of Systems, ed. L. Dekker.
pp. 429-438
3. O-J. Dahl, B. Myhrhaug, K. Nygaard: The SIMULA 67 Common Base Language, Norwegian Computing Center, 1970. S-22.
4. SIMULA Newsletter
Association of SIMULA Users - Norwegian Computing Center 1973-
5. Gáspár A. - Kocsis J. - Lamm P. - Visontay Gy.:
Az MTA tervezett számítógéphálózatának minőségvizsgálata.
Információ Elektronika 1978/4. 261-265.o.
6. A. Gáspár - P. Lamm : Simulation of packet-switched communication network
ACM SIGCOMM, Computer Communication Review.
Vol. 8. No.4. 1978. pp. 19-29.
7. Gáspár A. - Kocsis J. - Lamm P. - Visontay Gy.: Az MTA tervezett számítógéphálózatával kapcsolatos tervezési szempontokról
MTA SzTAKI, Tanulmányok 1978/87. 179-208.o.

8. Gáspár A.: Egy természetes számítási rendszer modellezése
MTA SZTAKI, Tanulmányok, 1973/12.121-145.o.
9. Gáspár A. - Pálvölgyi L. - Valló Á.: A neurokibernetika, a modellezés néhány metodológiai-filozófiai problémája.
ELTE TTK Filozófiai Közlemények 1975/II. 145-186.o.
10. A. Gáspár - L. Pálvölgyi: Zur Modellierung und Rechnersimulation des Lernens: ein Modellierungsversuch
Grundlagenstudien aus Kybernetik und Geisteswissenschaft. 1978/4. pp.113-125.
11. E. Holbaek Hanssen - P. Handlykken - K. Nygaard: SYSTEM DESCRIPTION AND THE DELTA LANGUAGE.
Norwegian Computing Center 1975. No.523.
12. Visontay Gy. - Csáki P.: DOCUM - a tool for documentation of SIMULA programs
SIMULA Newsletter 1979/2.
13. Gáspár A. - Visontay Gy. - Csáki P.: A CDC 3300 SIMULA/MASTER új eszközei felhasználói taskrendszerek számára
CDC Felhasználói Ismertető 11. 1977.május
14. K. Nygaard - O.J. Dahl: The Development of the SIMULA Languages.
ACM SIGPLAN Notices. Vol.13.No.8.August 1978.
15. O.J.Dahl. - E.W.Dijkstra - C.A.R. Hoare: Struktúrált programozás. Műszaki Könyvkiadó, 1978.

A SIMULA 67 UTÁNZÁSI NYELV ALKALMAZÁSA TELE-
FONRENDSZEREK FORGALMI VIZSGÁLATÁHOZ

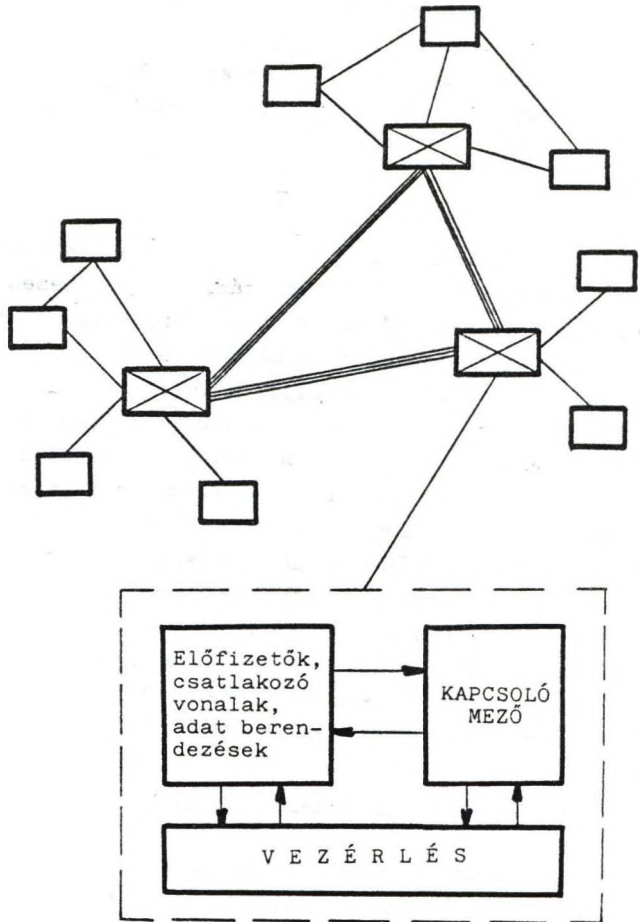
Ágostházi Margit * Gosztony Géza *
Nagy Rozália * Rétné, Fóti Mária *

Bevezetés

A BHG Híradástechnikai Vállalat kb. 1970 óta használja a SIMULA 67 utánzási nyelvet olyan számítással nehezen vagy egyáltalán nem elemezhető telefonrendszerek vizsgálatához, amelyekben az időviszonyok nyomkövetése elengedhetetlen. Az alábbiakban röviden ismertetjük a telefonforgalmi-méretezés főbb kérdéscsoportjait, majd példaként bemutatunk néhány olyan rendszert, amelyhez SIMULA 67 utánzó program készült. A vizsgált rendszereket részletesen felvázoljuk annak érdekében, hogy a megoldandó problémákról pontos képet adjunk és így a SIMULA 67 hatékonyságát is szemléltessük. Nem célunk átfogó képet adni az egyes utánzóprogramokról, sem pedig elemezni a kapott eredményeket. A programok részben az MTA-SZTAKI-val kötött szerződések keretében, részben a BHG Fejlesztési Intézetében készültek és CDC 3300 típusú számítógépen futtathatók.

1. Telefonrendszerek forgalmi kérdései

A telefonhálózat az üzenetkapcsoló rendszerek /pl. telex-és adathálózat, számítógép hálózat/ egyik jellegzetes esete. A ma már valóban világméretű telefonhálózat forgalomforrásokból és nyelőkiből /pl. telefonkészülékek, adatvégállomások stb./, kapcsolóközpontokból és az ezeket összekötő vonalnyalábokból áll. /l. ábra/. A kapcsolóközpontok rendszerint több hálózati síkban, hierarchikusan helyez-



1. ábra. A telefonrendszer egyszerűsített szerkezete

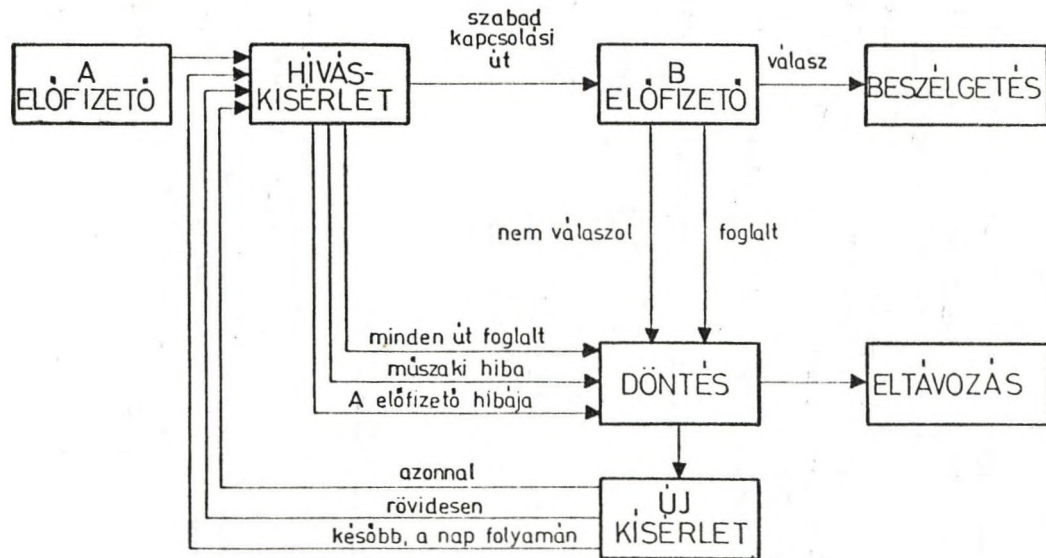
kednek el, a vonalnyalábok műszaki megvalósítása az előfizetői érpároktól a sokcsatornás vivőfrekvenciás vagy digitális elrendezésekig sokféle lehet.

A forgalom lebonyolítása szempontjából a telefonrendszerek tömegkiszolgálási rendszernek tekinthetők, a vizsgálatokban mind az előfizetők viselkedését /emberi tényező/, mind a kiszolgálás real time jellegét figyelembe kell venni. A forgalmi méretezés arra irányul, hogy a telefonrendszer összetevői mennyiségileg és a forgalom lebonyolítás szervezését tekintve megfelelőek legyenek, vagyis ne okozzanak kényelmetlenséget a felhasználóknak.

Vizsgálataink elsősorban kapcsolóközpontokra vonatkoznak, hiszen ezek a BHG termékei. A kapcsolóközpont vázlatosan három részre bontható /1. ábra/, ezek: a csatlakozó berendezések, az ezek összeköttetését megvalósító kapcsolómező és a működést irányító vezérlés. Ez a tevékenység-központú felosztás a műszaki megoldástól függően tartalmazhat átfedéseket, de napjaink kapcsolóközpontjaira általában alkalmazható.

2. Megismételt telefonhívások

Megismételt telefonhívások főleg túlterhelt telefonhálózatokban keletkeznek, az ismétlés jellegzetességeit és mértékét a hívó és hívott előfizetők viselkedése szabja meg, továbbá az, hogy a híváskisérletek a hálózatban milyen sikertelenségi okokkal találkoznak, [1]. A jelentéget a 2. ábra szemlélteti. Az elméleti vizsgálatot az nehezíti elsősorban, hogy a híváskisérletek érkezésére a rendszer állapota hatással van, visszacsatolás jön létre.



2. ábra. A hívásismétlés folyamata

A jelenség fontos paraméterei: a hívó kitartásfüggvénye /annak valószínűsége, hogy az x . sikertelen kísérlet után $/x+1/-$ edik kísérletet tesz /, a híváskísérletek közti szünetidő eloszlása, a sikertelenségi okok fennállási idejének eloszlása /pl. mennyi ideig foglalt a hívott/, stb. méreésekből ismertek, azonban az elméleti modellekben ezeket csak jelentősen egyszerűsítve lehet felhasználni. Külön nehézséget jelent az, hogy pl. a kitartásfüggvény vagy a szünetidő eloszlása függ a sikertelenség okától, 3. ábra.

Az utánczó program a különféle közelítő számítások ellenőrzésére készült azzal a végső céllal, hogy gyakorlati méretezési módszert alakítsunk ki.

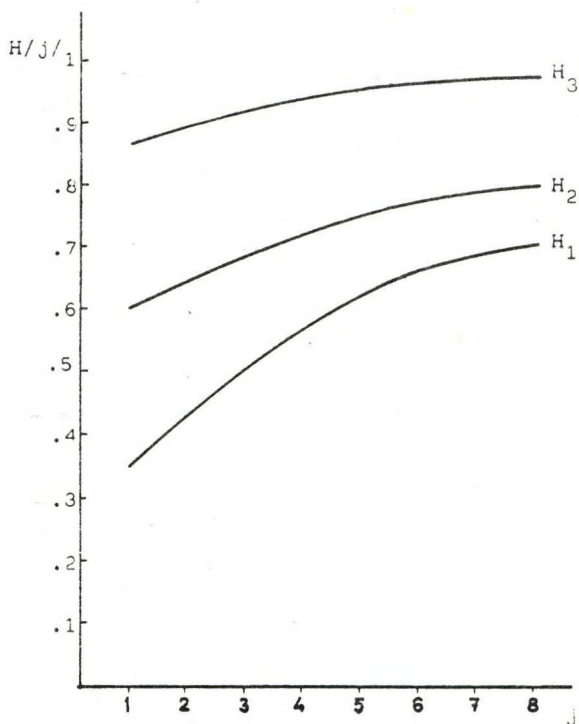
Az elkészült programokban egy- és kétfokozatos elrendezésű veszteséges és/vagy várakozásos kapcsolórendszereket modelleztünk. Az említett paramétereket szakaszos közelítő függvényekkel képeztük le [2,3].

A vizsgálat a Posta Kísérleti Intézettel szoros együttműködésben 1974 óta jelenleg is folyik. Az eddigi eredmények alapján, [4], méretezési segédlet készült a Magyar Posta számára [5, 6]. A megismételt telefonhívások jelenségének pontosabb ismerete mind a Magyar Posta, mind a BHG forgalmi méretezési gyakorlatában jól hasznosítható.

3. Kapcsolómező forgalmi méretezése

3.1 A kapcsolómezőről általában

A telefonközpontban a kapcsolómező biztosítja, hogy az előfizetőt a kívánt beszélgetés időtartamára egy másik előfizetőhöz kapcsolhassuk. Ha a hívott előfizető egy



3.ábra. Kitartásfüggvények /példa/

H_1 - nincs válasz

H_2 - a hívott foglalt

H_3 - torlódás

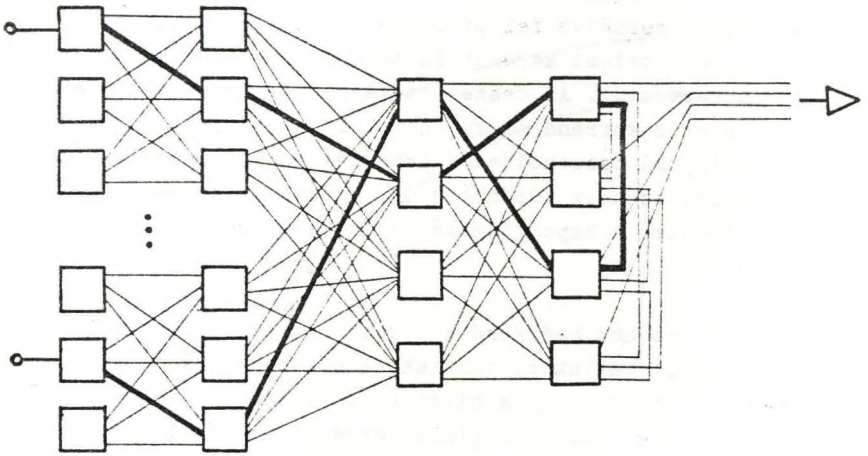
másik telefonközpontba van bekötve, akkor a kapcsolómező az előfizetőt a két központot összekötő csatlakozó vonalak valamelyikéhez kapcsolja.

Szerkezetileg a kapcsolómező kapcsolómátrixokból áll. Ezek ún. szerkezeti fokozatokat alkotnak. A szomszédos fokozatok mátrixait csatolóútak kötik össze. Két mátrix között tetszőleges számú csatolóút lehetséges. A kapcsolómezőben a beszédút felépítéséhez mindig 2 mátrix között kell összeköttetést keresni csatolóútakon és kapcsolómátrixokon keresztül. Az összeköttetéshez felhasználható elemeket és ezek sorrendjét külön meg lehet adni.

Kétféle típusú összeköttetés lehetséges: egyirányú és visszahurkolt, ezekhez különböző ún. szabad útkereső eljárások tartoznak. A kapcsolómező szerkezetét szemlélteti a 4. ábra.

A kapcsolómezőhöz beérkező hívások forgalomforrásokból /előfizetők, csatlakozó vonalak, stb./ érkeznek. Gyakran feltételezhetjük, hogy a hívások beérkezési intenzitása állandó, függetlenül a foglalt források számától /Erlang források, Poisson típusú bemeneti folyamat/. A hívásintenzitás lehet arányos az éppen szabad források számával /Engset források, a szabad forrás képvisel Poisson típusú bemeneti folyamatot/. Ha valamennyi forrás, ill. csatlakozó vonal foglalt, akkor a beérkező hívások intenzitása zérusra csökken.

Ha a kívánt összeköttetés nem valósítható meg azonnal, azaz torlódás lép fel, akkor vagy új kísérletet kell tenni /veszteséges rendszer/ vagy lehet várakozni a csatlakozás létrejöttéig /várakozásos rendszer/.



4.ábra. Többfokozatú csatolóútas kapcsolómező

3.2 Kapcsolómező utánzó programok

Kapcsolómezők forgalom áteresztő képességét a rendszer bonyolultsága miatt általában csak számítógépes utánzással lehet pontosan meghatározni. Az utánzóprogram lehetővé teszi a rendszer torlódási valószínűségeinek meghatározását, és a forgalmi viszonyok felmérését. Ezek ismeretében a forgalmi méretezése elvégezhető, ill. közelítő számítások ellenőrizhetők.

A BHG első kapcsolómező utánzóprogramját 1970-71-ben dolgozták ki. Ebben az ún. eseményhű utánzást használtuk és tetszőleges szerkezetű veszteséges és várakozásos rendszerek vizsgálatára alkalmas volt [2,7]. A program felhasználását az korlátozta, hogy a leképezhető kapcsolómező a gyakorlatban alkalmazott méretekhez képest kicsi volt.

Az 1977-78-ban készült programban a SIMULA gondolatvilágához közelebb álló időhű módszert alkalmaztuk. A kapcsolómező mintegy 1000-1500 bemenetes lehet három fokozatú mező esetében. A memória felhasználás a mindenkori mező nagyságán kívül a rendszer forgalmától is függ, melyre vonatkozóan részletes elemzéseket végeztünk [8].

3.3 Rendszerleírás a SIMULA 67-ben

A másodiknak említett program nagymértékben kihasználja a SIMULA 67 programnyelv adta lehetőségeket. Az alábbiakban vázlatosan felsoroljuk a nyelvből felhasznált elemeket:

4.1 Osztály, részosztály

4.2 BASICIO /INFILE, OUTFILE, DIRECTFILE/ osztály

- 4.3 SIMSET /LINK, HEAD/ osztály
- 4.4 SIMULATION /PROCESS, TIME, HOLD, CANCEL, ACTIVATE, REACTIVATE/ osztály
- 4.5 NEGEXP, RANDINT, DISCRETE véletlen szám generáló
- 4.6 SH, PROD, MOD bitműveletek végzésére szolgáló inline operátorok
- 4.7 Megszakításos programfuttatáshoz igen hatékonyan használja a program az 1977-ben a CDC 3300-ra kidolgozott RTLGENERATE, TASKPASSIVATE, TASKSAVE eljárásokat. Ezek segítségével egy futó program pillanatnyi állapota kimenthető, /a program futásának megszakításával vagy anélkül/ majd tetszőleges idő múlva újra indítható a futás a kimentett állapottól.

3.4 Alkalmazás

A régebbi program elsősorban az eseményhű utánzási módszer kiegészítésével kapcsolatos vizsgálatokat szolgált, továbbá felhasználtuk a BHG QA 96 típusú tárolt programvezérlésű /TPV/ alközpont forgalmi méretezésének kezdeti szakaszában. Az újabb programot üzemszerűen alkalmazzuk a fejlesztésben levő nagyobb kapacitású TPV alközpontok kapcsolómezejének vizsgálatához.

4. Tárolt programvezérlés forgalmi vizsgálata

4.1 A tárolt programvezérlés

Tárolt programvezérlés /TPV/ esetében a telefonközpontot számítógép vezérli. Az összeköttetések felépítése és lebontása során elvégzendő műveleteket a vezérlő számítógép programjai irányítják. Az igények a számítógép perifériáitól érkeznek, és a tennivalókat ugyancsak a perifériáknak

kiadott utasítások tartalmazzák. Perifériának számít a kapcsolómező, a csatlakozó vonalak, áramkörök, kezelőkészlet, stb. /l. ábra/. A vezérlő számítógép elláthat más feladatokat is, így pl. automatikus hibafelderítés és azonosítás, forgalom mérés, díjelszámolás.

A vezérlő software három részre osztható:

- a perifériákkal kapcsolatot tartó input-output programokra /pl. letapogatás, kapcsolómező-működtetés, stb./;
- az összeköttetések felépítését, lebontását irányító, bonyolult logikai feladatokat végző, ún. funkcionális programokra, és
- az előbbiekből felépülő programrendszert szervező monitor-programra.

A forgalmi méretezés szempontjából a TPV software a kiszolgálószervek /programok/ halmazának tekinthető. A hívás típusától /házi-, helyi- vagy távhívás/ függően más-más kiszolgálószervekre van szükség a hívásfeldolgozás során, és ezek igénybevételeinek sorrendje szintén meghatározott. A kívánt összeköttetés felépítése és lebontása kiszolgálási igények sorozataként jelenik meg és a következő igényt az előző igény kiszolgálása, vagy a perifériáktól érkező jelzés /pl. állapotváltozás/ váltja ki. Mivel a vezérlő programrendszer egyidejűleg több hívás párhuzamos feldolgozásával foglalkozik az egyes programok előtt az igények várakozási sorokat alkotnak, és meghatározott szabályok szerint kerülnek kiszolgálásra.

A hívásfeldolgozás tehát a típusának megfelelő programok egymás után történő végrehajtása, miközben a külvilágból a perifériák közvetítésével érkező jelzések /pl.

jelentkezés, számjegybeadás/, a programrendszer pillanatnyi állapota /pl. terhelés, monitor-ciklus fázisa/, és külső feltételek /pl. torlódás, hívott foglalt/ bonyolult kölcsönhatásba lépnek egymással.

A programok közül van olyan, amelynek lefuttatása bizonyos halasztást tűr, de vannak olyanok, amelyek sürgősek és késésük káros következményekkel pl. információ veszteséssel jár /ilyen pl. a számjegybevételezés/. A vezérlő programrendszer akkor jut el forgalmi kapacitásának határára, ha teendőt olyan késésekkel, vagy veszteségekkel képes csak el látni, hogy az már zavarja a felhasználókat.

A TPV forgalmi méretezésének általános kérdéseiről pl. [9] ad általános tájékoztatást.

4.2 Az utánczó program alapelve

Egy vezérlő programrendszer kifejlesztésének legkülönbözőbb fázisaiban merül fel az az igény, hogy a készülő rendszer forgalmi jellemzőit megbecsüljük. Az előzőekben vizsgolt bonyolult várakozásos rendszer tisztán matematikai elemzése többnyire elháríthatatlan nehézségekbe ütközik, ezért a kérdés megoldásához utánczásra van szükség.

A kidolgozott utánczási módszer lényege az EPR /Call Event and Program Request Chain/, azaz hívás események és az igényelt programok láncolata. Minden hívás tipushoz más-más EPR tartozik, és ez kombinációja pl. az előfizető viselkedésének vagy a csatlakozó telefonközpontból jövő jelzések sorozatának és a vezérlő software iránti igényeknek. Az 5. ábrán az EPR-ben foglalt információ egyszerűsített példája látható.

A módszer fő előnye az, hogy a teljes software-tervezési folyamat során azonos alapelven felépülő és azonos szerkezetű utánczó-program alkalmazható. Az EPR a szükséges mértékben lehet vázlatos vagy részletes és így a pillanatnyi igényektől függően megvalósítható a durva közelítés, a folyamatot egymástól független részekre bontó /ún. subcall típusú/ utánczás vagy a folyamat minden összefüggését leképező /ún. call típusú/ utánczás.

4.3 Az utánczó program szerkezete

A SIMULA nyelv adottságait messzemenően kihasználó és az adott felhasználáshoz további software eszközökkel kiegészített programrendszer [10] három alapvető részből épül fel.

- Az EPR-ek

Egy EPR a hívás típusok egyikére vonatkozik, és magában foglalja

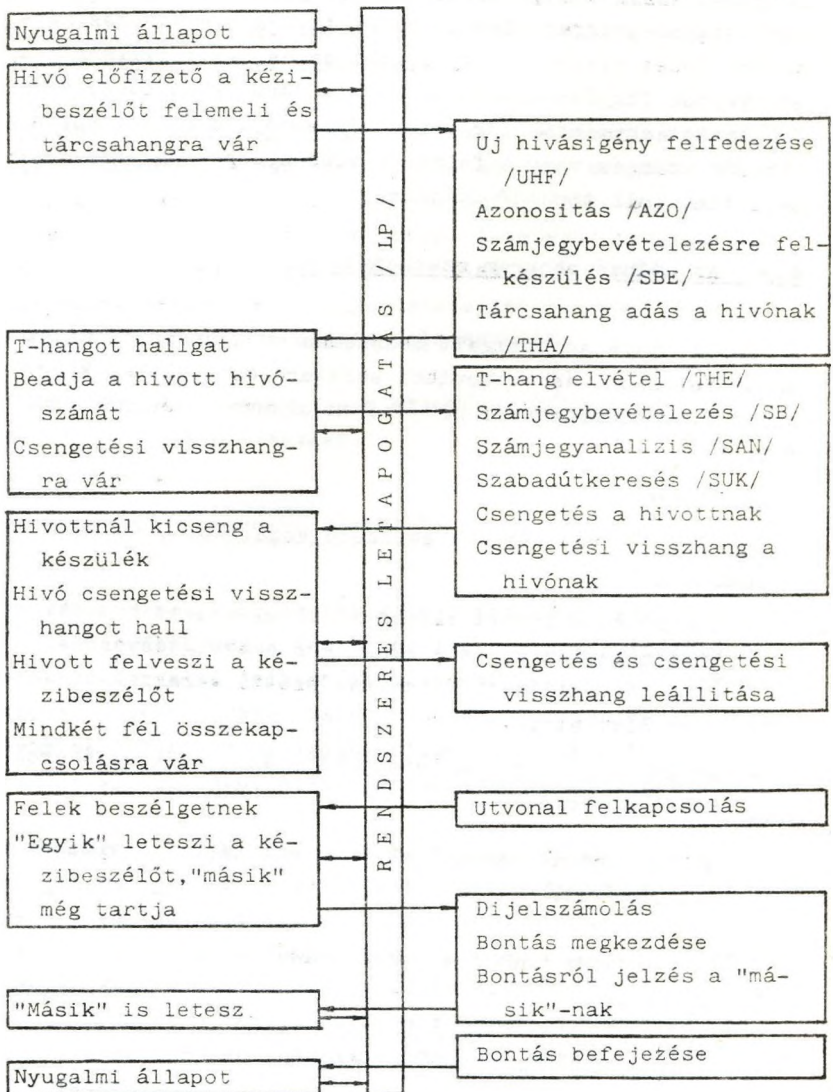
- a hívó és a hívott előfizető által kezdeményezett minden eseményt, és a szükséges emberi tényezőket /pl. tartásidőket, tárcsázás előtti várakozást, válaszidőt, stb./;
- az adott hívástípus lebonyolításához szükséges programok sorrendjét;
- a TPV rendszer modelljében le nem képzett rendszerjellemzőket.

- A TPV rendszer modellje tartalmazza a

- monitort,
- hívásfeldolgozó programokat,
- társakat és más hardware-elemeket.

ELŐFIZETŐ VISELKEDÉSE

VEZÉRLŐ SOFTWARE IRÁNTI IGÉNY



5.ábra. EPR /példa/

A ciklus- és prioritásszervezésért felelős monitor részletesen le van képezve. A programokat egy vagy több állapotfüggő tartásidővel rendelkező kiszolgálószerveknek tekintjük. A tárolók szerkezetét, stb. csak akkor képezzük le, ha befolyásolják a hívásfeldolgozást.

- Kiértékelő rutinok

A kiértékelés az alábbiakra vonatkozik:

- az EPR-ben leirt események közötti idő-intervallumok /makroidók/;
- programokon belüli időintervallumok /mikroidók/;
- különböző programok előtti várakozási sorok;
- foglalt tárolók és egyéb eszközök.

Az EPR szemléltetésére bemutatunk egy részletet, amely az 5. ábrán látható folyamat egyik részét ábrázolja.

•
•
•
HOLD /nyugalmi állapot/
VAR /LP/
VAR /UHF/
VAR /AZO/
VAR /SBE/
VAR /THA/
HOLD /T-hang hallgatás/
VAR /THE/
HOLD /Számjegybeadás/
•
•

A VAR /H/ eljárás /amelyet az EPR-ben deklarálunk/ az EPR obejtumok egy példányát beállítja az igényláncban soronkö-

vetkező H funkcionális program sorába, és az EPR ott vára-
kozik a kiszolgálás befejezéséig, pontosabban a H-től ér-
kező visszajelzésig. A funkcionális programok a sorból az
érvényes rend /FIFO, LIFO, RANDOM/ veszik sorra az igénye-
ket és visszajeleznek az EPR-nek. A szövegesen jelzett idő-
tartamok hosszúságai az előfizető tevékenységétől függnek,
ezeket többnyire valószínűségi eloszlások formájában tar-
talmazza a program.

4.4 Alkalmazás

Az ismertetett típusú programokat több telefonközponti
vezérlő programrendszer forgalmi vizsgálatához alkalmaz-
tuk.

5. Köszönetnyilvánítás

A szerzők köszönetüket fejezik ki az ismertetett progra-
mok készítésében közreműködött kollégáknak: Szentirmai
Ferencnének /BHG-FI/, Szádeczky Kardoss Tamásnak /BHG-FI/
Jantó Istvánnénak /BHG-FI/ és Szabó Gábornak /MTA KFKI/.

Köszönet illeti az MTA SZTAKI mindazon munkatársait, első-
sorban Knuth Elődöt, akik mind a szerződéses munkák kere-
tében, mind a software tanácsadás során jelentős támoga-
tást adtak feladataink elvégzéséhez.

Irodalom

- [1] Ágostházi M., Gosztony G.: A megismételt hívások
jellemző vonásai.
Híradástechnika, 26. 1975. 4. 109-119.
- [2] Knuth, E.: Telefonközpont működésének szimulációja.
Számítógépes rendszerszimuláció szimpózium, Budapest,
1975. pp. 23-42.

- [3] Gosztony, G., Szentirmai, F.: Számítógépi utánzóprogramok megismételt hívások hatásának utánzásához. BHG-FI V-1/78 sz. jelentés.
- [4] Gosztony, G.: Comparison of calculated and simulated results for trunk groups with repeated attempts. 8. ITC, Melbourne, 1976. Paper 321. pp. 1-11.
- [5] Ágostházi, M., Gosztony, G., Honi, G., Nagy, R.: Méretezési segédlet megismételt telefonhívásokhoz. 8. Magyar Operációkut. Konf. Szeged, 1978. Előadáskivonatok C2/2 pp. 117-122.
- [6] Ágostházi, M., Gosztony, G., Honi, G.: A megismételt hívások hatását figyelembe vevő gyakorlati méretezési módszer. "100 éves a bolgár távközlés" konferencia, Szófia, 1979. Előadás. /Megjelenés alatt PKI Közlemények-ben/
- [7] Rétné, Fót M.: Telefonközpontok forgalmának utánzása számítógép segítségével. BME doktori disszertáció, Bp. 1971. p. 87.
- [8] Nagy, R.: Számítógépes utánzóprogram kapcsolómező forgalmi méretezéséhez. BHG-FI. 90110-159/1978 sz. jelentés
- [9] Villar, J.E.: Traffic calculations in SPC systems. 8. ITC, Melbourne, 1976, Paper 611, pp. 1-7.
- [10] Knuth, E., Sárközi, A.: Diszkrét vezérlési rendszerek modellezése. Rendszerelmélet'79 konferencia, Sopron, 1979. Előadás.

Az IGYR-630 integrált gyártórendszer szimulációja

Kun István

Az IGYR-630 integrált gyártórendszer a Csepel Vas- és Fémművek Szerszámgépgyárában kerül kifejlesztésre, az MTA Számítástechnikai és Automatizálási Kutatóintézet és a Gépipari Technológiai Intézet közreműködésével. A rendszerben emberi kéz érintése nélkül általában kis szériákban bonyolult alkatrészek készülnek.

A rendszernek a szimulációs programban figyelembevett része 4 mikroprocesszorral vezérelt forgácsológépből, egy munkadarabszállító automatából, a megmunkálási fázisok /szám szerint 4-8/ közötti tárolásra szolgáló pufferhelyekből, valamint a folyamatirányítást végző TPA kiszámítógépből áll.

Az "integrált gyártórendszer" kifejezés azt jelenti, hogy az előbbi /és egyéb/ fizikai komponensekhez megfelelő software is társul, amely lehetővé teszi, hogy közvetlen emberi beavatkozásra csak kivételes esetekben /pl. üzembiztonság vagy tervmódosítás/ legyen szükség.

Az integrált gyártórendszer termelésirányításának célja az, hogy a termelési határidőket figyelembevéve, a negyedévi tervet több lépésben lebontva, műszakonként optimális menetrendet állapítson meg. Az optimalitás itt főleg a kapacitáskihasználásra vonatkozik.

A szimulációs program elkészítését főleg a termelésirányítási programrendszer kidolgozása tette szükségessé. Egyrészt, amíg a fizikai rendszer nem működik, addig a termelésirányítási programok működőképességének kipróbálása kizárólag szimulációval lehetséges.

muláció útján történhet. Másrészt a fizikai rendszer beindulása után a további kísérleteket és finomításokat összehasonlíthatatlanul olcsóbb egy számítógépes programon kipróbálni, mint valóságos munkadarabokon. Természetesen a szimuláció nemcsak egy rögzített paraméterekkel rendelkező rendszert képes helyettesíteni a termelésirányítási software számára, hanem ki lehet rajta próbálni a paraméterek megváltoztatásának hatását is. /Pl. más összetételű gépcsoport, más jellegű munkadarabok, stb./

A szimulációs program elkészítését lényegesen könnyítette a SIMULA 67 programozási nyelv "class" /osztály/ koncepciója. Ennek segítségével ugyanis lehetővé és főleg áttekinthetővé vált a munkadarabok és megmunkáló gépek absztrakt leírása.

A program tulajdonképpen kétszintű. Az alsó szinten azok az elemek és logikai kapcsolatok szerepelnek, amelyek valamennyi hasonló jellegű integrált gyártórendszerben közösek. Minden munkadarab a számára előírt technológiai sorrendben végigjárja a megmunkálást végző gépeket, és azokon az ugyancsak a technológia által megszabott időt tölti. Ha pedig a következő megmunkálási fázist végző gép foglalt, akkor vár, amíg az illető gép felszabadul, majd pedig a folyamatirányítást reprezentáló programrész eldönti, hogy az adott gépre várakozó munkadarabok közül melyik kerüljön megmunkálásra. A megmunkálógépek maximális leterhelése ebben a felfogásban egy ütemezési /job-shop scheduling/ probléma. Az optimumot biztosító egzakt algoritmus csak két gép esetére ismert, az adott rendszer négy gépénél valamilyen heurisztikus algoritmus kidolgozása a cél.

Ezt a szimulációs program a következőképpen segíti elő: inputként a szóbajövő munkadarabhalmazt és a munkadarabcsere döntési módszerét kapja, outputként pedig egy néhányórás időszak lefolyását ábrázoló Gant-diagramot és az egyes megmunkálógépek üresjáratú idejének arányát adja. Így a munkadarab-összetétel és/vagy a döntési kritérium megváltozásának hatása szemléletesen nyomonkövethető.

A program alsó szintje nem veszi figyelembe a munkadarabszállítás által okozott idővesztéséget, mert ez minden konkrét fizikai rendszernél másképp alakul. A program felső szintje az, amely a csepeli IGYR-630 munkadarabmozgatási mechanizmusát leírja. Egy sínen mozgó kocsi szállítja a gépek és a puffertárolók között a munkadarabokat, ez a kocsi a ki- és berakodásra is képes. A kocsi feladatai közé tartozik az is, hogy az egyes munkadarabokat a következő megmunkálási fázist végző géphez csoportosítsa. Az utóbbi tevékenység azonban bármikor megszakítható, mert a megmunkálógépek folyamatos munkadarabellátása prioritást élvez. Megszakítás esetén a kocsin található munkadarabot a legközelebbi szabad puffertárolóba kell elhelyezni.

A program alsó szintje a szállítás szimulációja nélkül önállóan is működőképes. A szállítás tulajdonképpen csak szuperponálódik az általános tevékenységekre, bonyolultságát jellemzi azonban, hogy általa a program hossza kb. a kétszeresére nő.

Az ismertetett példa jól illusztrálja, hogy egy szimulációs program elkészítése mennyire igényli a rendszerelméleti megközelítést, és hogy az utóbbit mennyire megkönnyíti a SIMULA 67 nyelv fogalomrendszere.

A közúti forgalom szimulációs vizsgálatának előnyei

dr Vásárhelyi Boldizsár, a közl.tud. kandidátusa

A közúti forgalom igen bonyolult történés, ahol a véletlenszerűségnek is nagy szerepe van. Véletlenszerű például egyrészt az, hogy az egyes járművek egy adott ponton mikor haladnak át, milyen a sebességük, a járművezetők sietnek-e vagy ráérnek, erőszakosan vezetnek-e vagy nyugodtan, másrészt pedig az, hogy hogyan hatnak egymásra a forgalom résztvevői.

A különböző közlekedési ágak közül a közútra a legjellemzőbb a véletlenszerűség mindkét szempontból. A közúti közlekedésben vesz részt a legtömegesebben a nagyközönség. A magán járművezetőket még tekinthetjük betanított munkásoknak, de gyalogosként mindenki közlekedik, akár részesült közlekedési oktatásban, akár nem.

A közúton a legtöbb járművezető saját döntéseinek megfelelően, menetrendi kötöttségek nélkül halad a szűkösen kétdimenziós pályafelületen. Az egydimenziós vasúton szoros a menetrendszerűség és a forgalomirányítás, a kétdimenziós vízfelületeken, illetve a háromdimenziós légtérben általában nagyobb hely áll a járműveknek a manőverezéshez rendelkezésre, a kritikus helyeken pedig szoros külső irányítást kapnak a többségükben magasan kvalifikált járművezetők.

Ugyanakkor az uttervezőknek, üzemeltetőknek is szükségük van a közúti "forgalmi folyamat" törvényszerűségeinek minél alaposabb ismeretére. Ismerni kell például, hogy

- az egyes uttípusoknak mekkora a kapacitásuk /hány jármű haladhat át rajtuk óránként/
- különböző forgalomnagyságok és forgalomösszetételek mellett milyenek a forgalmi körülmények /átlagsebesség, előzési lehetőségek, stb./

- a csomópontokon mekkora a keresztezni vagy becsatlakozni tudó forgalomnagyság,
- milyen nagy forgalomnál kell jelzőtáblás vagy jelzőlámpás szabályozást bevezetni a csomópontokon,
- különböző esetekben milyen forgalomszabályozási intézkedések a legjobbak,
- az utak hiányosságai, illetve a közelműnyezet hatásai /szűköletek, éles ívek, emelkedők, kilátásbeli akadályok/ hogyan befolyásolják a forgalmat, milyen költségtöbbleteket okoznak,
- hogyan lehet a forgalom körülményeit azonnal figyelembe vevő forgalomirányítást jól végezni,
- esetleges jövőbeni változások esetén milyen következményekkel kell számolni.

A kutatók régóta foglalkoznak a közúti forgalom vizsgálásával és számos matematikai modellt dolgoztak ki. Mint ismeretes, ezek egy csoportja determinisztikus, azaz nem veszi figyelembe a véletlenszerűséget, hanem például fizikai, áramlástan analógiából indul ki. A másik modellcsoport sztochasztikus, azaz a véletlenszerűségből indul ki és a valószínűségszámítás eszközeivel közelíti meg a kérdést. A probléma viszont az, hogy ezek a modellek nem tudják kielégítően jellemezni a gyakorlatban előforduló forgalmi helyzeteket, mert vagy erőltetett analógiákat, vagy túlzott egyszerűsítéseket alkalmaznak, így hibás eredményre vezetnek, ellenkező esetben pedig kezelhetetlenné válnak. A valószínűségi modelleknél gyakori függetlenségi feltételek itt nem felelnek meg a tényleges helyzetnek.

Hasonlóan számos tudományos és gyakorlati szakterülethez, ahol a sztochasztikus szimulációs módszerek e téren is nagy jelentőségre tettek szert. Segítségükkel a vizsgált rendszert /pl. egy utszakasz a járműekkel/ elemeire bontjuk és a rend-

szerben lezajló folyamatokat a számítógép segítségével utnázuk. Eközben figyelemmel kísérjük az eseményeket be- és kilépés, haladás, manőverezés, az elemek közötti kölcsönhatásokat és gondoskodunk arról, hogy a kölcsönhatások a valóságnak megfelelően érvényesüljenek a folyamatok utánzásakor. Ugyanakkor messzemenően tekintettel lehetünk a környezet milyenségére is. A véletlenszerűséget egyrészt az egyes elemek tulajdonságainak "kisorsolásánál" /esetünkben: járműnem, sebességi igény, gyorsítóképeség, vezetési stílus, távolságbecslő képeség, biztonsági igény/, másrészt a rendszerbeli "viselkedésnél" is /esetünkben: mikor lép be a rendszerbe, hogyan manőverezik egy adott helyzetben/ vesszük figyelembe.

A használható eredményekhez jutás feltétele, hogy részmodellekben szereplő paramétereken /pl. a manőverezés jellemzők mérőszámai/ a tényleges helyzetet tükrözték. Ugyanakkor elképzelt vagy jövőben várható eseteket is szimulálhatunk /pl. "minden járművezető nyugodt", "automatikus távolságtartás"/ és hasonlíthatunk össze a ténylegesekkel.

Ezen a téren nagy szükség van forgalomtechnikai mérésekre, matematikai részmodellekre, és egyéb /pl. vezetéspszichológiai, járműdinamikai/ jellemzők megfelelően számszerűsített leírására.

A szimuláció során számos igen jól jellemző információ keletkezik a rendszer működéséről, amelyet össze kell gyűjteni. Ilyenek lehetnek a sebességeloszlások és átlagsebességek, a késleltetési és/vagy várakozási időveszteségek, az előzések száma, a veszélyes helyzetek száma, stb.

A rendszer és az elemek leírását, valamint a fentemlített közlekedési szabályokat a program és az input adatok tartalmazzák. Az eredményt tartalmazó információkat általában számértékek formájában kapjuk, de megfelelő berendezésekkel azok ki is rajzoltathatók.

Az előzőkkel összhangban a szimulációval nyerhető eredmények^{sok} információt tartalmaznak. Sok matematikai modell csak egyes átlagértékeket szolgáltat, míg a szimuláció a minket érdeklő eloszlásokat is megadja, azokat is, amelyeket matematikai modellekkel - ilyen mennyiségben - gyakorlatilag nem kaphatnánk meg.

A matematikai kérdésekre itt nem térnek ki. Az általában használt programnyelveket kiterjedten használják a közúti forgalom szimulálására, de a speciális szimulációs nyelvek /pl. SIMULA, GPSS/ is alkalmasnak bizonyultak.

A közúti forgalom szimulációjánál az időlépéses modellt szokás alkalmazni, másodperces, vagy rövidebb lépésközzel./A másodperces lépésköz egyes esetekben túl durvának bizonyul./ Ugyanis egy csomópontban vagy egy úton szinte mindig "történik valami", amit csak a rendszer állandó figyelésével lehet megbízhatóan nyomonkövetni és utánozni. A szimulált időszak elején levő u.n. "bemelegedési időt" figyelmen kívül szoktuk hagyni és csak a stabilizálódott folyamat eredményeit értékeljük.

A vonalas közúti létesítményeken haladó forgalom jellegzetes történése az előzés. Ennek vizsgálata fontos, mert az előzési lehetőség a forgalom minőségét behatóan jellemzi, ugyanakkor - különösen egypályás ellenirányu forgalmu utnál - balesetveszélyes manőver. Az előzéseknek a szimulációs technika szempontjából fontos azon következményük, hogy az úton haladó járművek sorrendje állandóan változik. Biztosítani kell viszont, hogy mindenegyed járművet végigkísérjünk a szimulációs modellben, mert a járművek viszik magukkal egyéniségüket és a többiekkel való kölcsönhatásba lépésük mikéntjét is.

A járművek azonosítása történhet úgy, hogy minden scanning végén átszámozzuk azokat az úton elfoglalt helyük sorrendjében. Másik lehetőség az ut /pl. 1 m-s vagy 1 járműhossznyi/ térközökre osztása és ezeken egy foglaltsági függ-

vény definiálása.

E függvény értéke: $\begin{cases} 0, & \text{ha nincs jármű a térközön,} \\ K, & \text{ha azon a } k\text{-ik jármű tartózkodik.} \end{cases}$

Mindkét esetben egyszerűen megkereshetők a vizsgált jármű mozgását befolyásoló előtte, utána, vele szembe, illetve a szomszéd sávon vele párhuzamosan haladó járművek, kölcsönhatásaik felmérhetők és követhető mozgásuk.

A közuti forgalom szimulációs modelljének "lelke" természetesen a forgalomban résztvevők viselkedésének utánzása. Az 1. ábra erre mutat egy egyszerű példát. Itt autópályáról van szó, a kétsávu pályán egyirányban haladnak a járművek, a külső "1" sáv szolgál a folyamatos haladásra, a belső "2" sáv az előzésre. A program mindegyik másodpercben minden járművet megvizsgál és előírja számára a viselkedést a következő időpontig. Az ábrán a "J" jelű járműre került sor.

A közuti forgalmi folyam szimulációját már sokféle feladattípusra elvégezték. A kérdésnek nagy irodalma van, itt csak saját vizsgálatainkból említünk kettőt.

Az utviszonyokat figyelembe vevő modellel meg lehetett állapítani, hogy a leggyakoribb típusu, kétsávu utak kanyargóssága és a rossz előrelátási viszonyok az átlagsebességet az ideális egyenes uthoz képest 5-15 km/órával csökkentik. Az egyenes utakon 2-5-ször annyi előzést lehet elvégezni, mint a kanyargósakon. Ezek az eredmények lehetővé teszik, hogy pl. az utkorszerűsítések gazdasági értékelésénél az ívek, látótávolságok javításának hatását jobb közelítéssel lehessen számitásba venni.

Megemlítjük, hogy ugyanilyen típusu modellel sikerült a hazánkban és Közép-Európában eddig nem épített, de több latin országban és Angliában kedvelt háromsávu ut forgalmát szimulálni és különböző forgalomösszetételek mellett össze-

vetni a kétsávu utra kapott hasonló eredményekkel. Ezen utak létesítése olyan kérdés, amelyben igen megoszlik a szakterület véleménye. Reméljük, hogy a fentiekből látható, hogy a szimulációs eljárások a jobb, gazdaságosabb, biztonságosabb forgalomlebonylódást lehetővé tevő forgalomtervezés, szabályozás és üzemeltetés fontossegédeszközét képezik. A gondolatmenetek egyéb közlekedési ágak vizsgálatánál is hasznosak lehetnek.

Irodalom

- Vásárhelyi B.: Szimulációs keretmodell ideális pályák forgalomlefolyasának vizsgálatára
/Közlekedéstudományi Szemle, 1974/7. 321-326.o./
- Brannolte, U. - Vásárhelyi, B.: Simulationsmodell für den Verkehrsablauf auf kurvigen Landstrassen
/Vorl. Bericht Nr. 22., Ifv., Universität Karlsruhe/
- Brannolte, U. - Vásárhelyi, B.: Kétsávu, síkvidéki, nem ideális vegyesforgalmu utak forgalmi folyamának szimulációja
/Közlekedéstudományi Szemle, 1978/12./

Diszkrét idejű sztochasztikus szabályozott rendszerek szimulációja a SIMULA nyelv segítségével

Hangos Katalin^{*}

Bevezetés

Az előadásban egy olyan szimulációs programrendszert mutatunk be, amelyet számítógépes szabályozó algoritmusok fejlesztése, valamint a szabályozott rendszerek vizsgálata céljából dolgoztunk ki. Miután digitális számítógépeken diszkrét idejű szabályozóalgoritmusok valósíthatók meg egyszerűen, ezért szabályozandó rendszereinket is diszkrét idejűnek választottuk; ezek lehetnek valóban diszkrét idejűek /pl. Åström-féle diszkrét lineáris rendszermodellel [1] leírhatók/ vagy folytonos idejűek, amelyek bemenő és kimenő jeleit mintavételező szervekkel mintavételezzük [2]. Mivel a szabályozók szempontjából a rendszer bemeneteinek és kimeneteinek összefüggései érdekesek, ezért a szabályozandó rendszert bemenet-kimenet modellel leírható alakúnak feltételezzük /nem zárva ki ezzel azt sem, hogy - bemenetek és kimenetek az állapotvektoron keresztül legyenek összefüggésbe hozhatók/.

A program felépítésénél a széleskörű felhasználhatóság érdekében a modularitást, a program részeinek könnyű cserélhetőségét tartottuk szem előtt. Ennek érdekében különválasztottuk a szabályozott rendszer elemeinek /szabályozott szakasz, szabályozók, identifikációt végző egységek/ azon funkcióit, amelyek az egymással való megfelelő kapcsolódást biztosítják a belső működés funkcióitól. Ez utóbbiakat - a SIMULA nyelv hierarchikus osztályszerkezetét felhasználva - a kapcsolódást biztosító osztályok részosztályaiként irtuk le. Így a részosztályok kicserélésével /ami az ún. opcionális editálás segítségével technikailag könnyen kivitelezhető/ lehetőség van a rendszerelemek belső szerkezetének egyszerű megváltoztatására. A rendszerelemek funk-

^{*}MTA Számítástechnikai és Automatizálási Kutató Intézet, Budapest

cióinak hasonló szétválasztását szakaszos és folyamatos technológiai elemeket tartalmazó vegyipari rendszerek szimulációja során is alkalmazták [3].

A szimulációs programrendszer tartalmazza a rendszerelemek kapcsolatát leíró alapegységek /szabályozott szakasz, szabályozó, identifikáló egység/ prototípusait és az elemek belső működését leíró prototípus-készletet, amely a felhasználó által tovább bővíthető. A prototípusok felhasználásával a SIMULA nyelvben akárhány példány generálható, különböző paraméterekkel.

A szimulációs programrendszert a szimuláció eredményeinek könnyebb áttekinthetősége és értékelhetősége érdekében statisztikai adatgyűjtő és rajzoló egységek prototípusaival egészítettük ki.

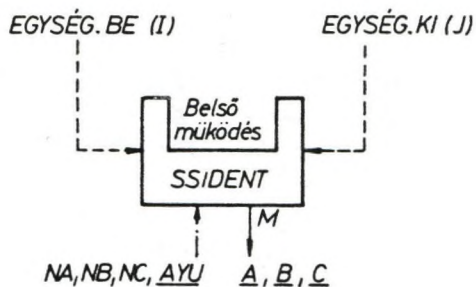
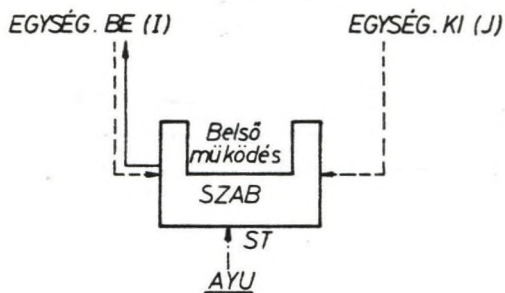
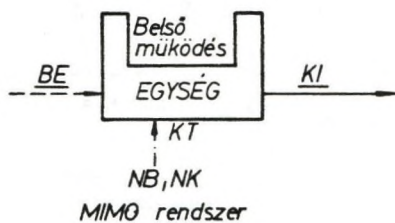
A továbbiakban az alapegységeket, majd az elemek belső működését leíró készletet ismertetjük, végül példákön mutatjuk be, hogy hogyan használható a program különböző szabályozott rendszerek vizsgálatára.

Az alapegységek

Az alapegységek sematikus rajza az 1. ábrán látható. Az ábrákon a rendszerelem /fizikai és információ/ bemeneteit szagatott, a működéséhez szükséges meghatározó paramétereket eredmény, a kimeneteket folytonos vonallal ábrázoltuk.

A valóságos viszonyokból kiindulva alapvető, más rendszerelemek nélkül is szimulálható alapegységnek a szabályozandó objektumot /EGYSÉG/ választottuk. Ez egy több-bemenetű több-kimenetű /a szabályozáselméletben megszokott rövidítéssel MIMO/ rendszer, amely saját óróval /KT/ rendelkezik és a belső működéstől, valamint az időtől függő módon a megadott számú /NB/ bemeneti értékekből /BE/ a megadott számú /NK/ kimeneti értéket állítja elő /KI/.

A szabályozó alapegység /SZAB/ egy egybemenetű-egykimenetű /SISO rendszer szabályozására képes. Egy megadott EGYSÉG-példány adott sorszámú /J/ kimenetének és adott sorszámú bemenetének /I/ értékeit figyelembe véve, a belső működésének



1. ábra
Az alapegységek




megfelelő módon állapítja meg az adott időpontbeli beavatkozó értéket. A szabályozó alapegységnek is saját ideje van /ST/, amely lehetővé tesz az egységhez képest aszinkron működést /későbbi bekapcsolást, ritkább beavatkozási kibocsátást/. Az alapegység működéséhez felhasználja az előre megadott alapjel értékeket /AYU/.

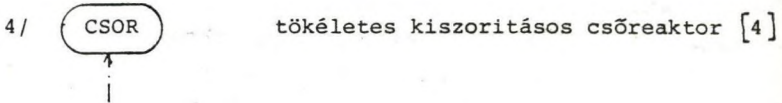
Az identifikáló alapegység /SSIDENT/ meglehetősen speciális: egy megadott EGYSEG-példány adott sorszámú /J/ kimenete és adott sorszámú /I/ bemenete között Åström-típusú lineáris modellt feltételez és ennek A, B, C együttthatóit becsüli a belső működéssel meghatározott módszerrel. A működéshez az előre megadott alapjeleket /AYU/ és az együtttható-polinomok fokszámait /NA, NB, NC/ használja fel.

Belső működés készletek

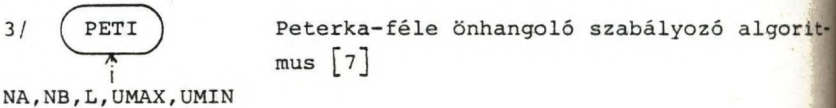
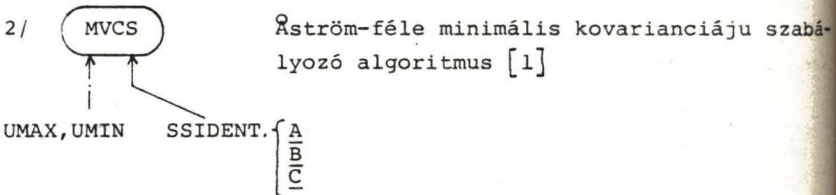
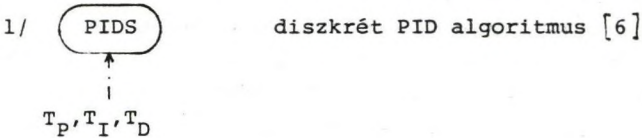
Az egyes alapegységek belső működéseinek készleteit szimulációs céljainknak megfelelően alakítottuk ki, új feladatok megoldásához a készlet egyszerűen tovább bővíthető. Nem célunk az egyes eljárások leírása, bővebbet az érdeklődő a megadott irodalmi hivatkozásokban talál. Itt csupán szemléltetésképpen a jelenleg rendelkezésünkre álló belső működések felsorolását adjuk:

Az EGYSEG alapegység belső működéseinek készlete:

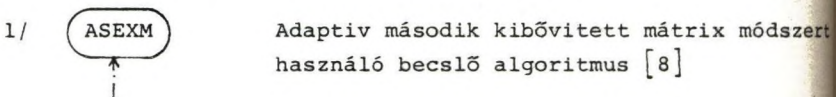
- | | | |
|----|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 1/ |  | diszkrét, lineáris, Åström-féle SISO rendszermodellel leírt rendszer [1] |
| 2/ |  | tökéletesen kevert tankreaktor [4] |
| 3/ |  | ARMA-folyamat szerint változó együttthatóju diszkrét, lineáris, Åström-típusú rendszer [5] |



A SZAB alapegység belső működéseinek készlete:

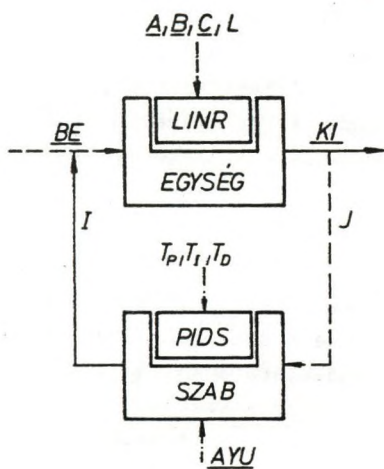


Az SSIDENT alapegység belső működéseinek készlete:



Példák

A 2. ábra egy egyszerű szabályozott rendszer - egy egybemenetű-egykimenetű lineáris Åström-moddellel leírható rendszer PID szabályozóval - sematikus rajza.



2. ábra

Egy egyszerű szabályzott rendszer

A rajzoló és statisztikai adatgyűjtő egységek ábrázolását /amelyek madjñem minden konkrét szimulációs program részei/, az egyszerűség kedvéért elhagytuk.

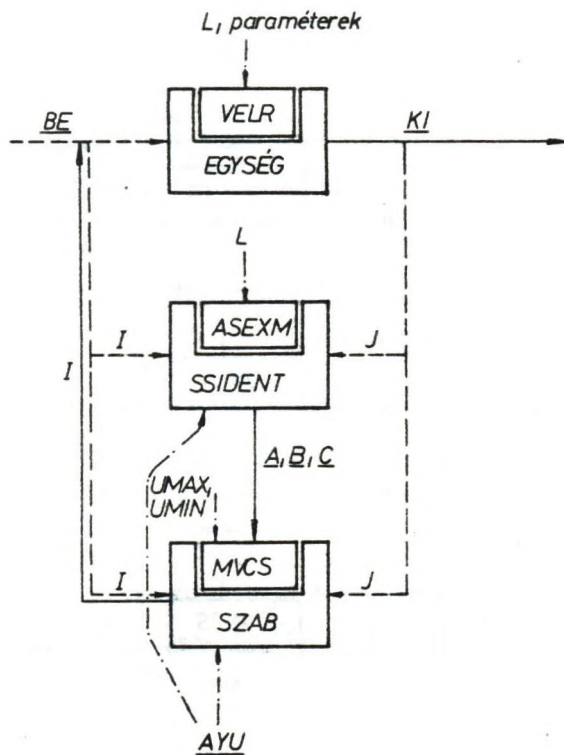
Kissé komplikáltabb a 3. ábrán látható szabályozott rendszer. Elemei egy ARMA folyamat szerint változó együtthatós egybemenetű-egykimenetű diszkrét lineáris Åström-modellel leírható rendszer, egy Åström-féle minimális variancájú szabályozóval, amelynek együtthatóit az identifikáló egység által meghatározott rendszerparamétereiből számoljuk /az identifikáló egység hangolja a szabályozót/.

A 4. ábrán olyan rendszerre mutatunk példát, ahol egy több-bemenetű több-kimenetű rendszert több szabályozóval szabályozunk. Az ábrabeli példában a két szabályozókör és ezáltal a két különböző típusú /Peterka-féle önhangoló, ill. PID/ szabályozó kölcsönhatását elhanyagoltuk.

Összefoglalás

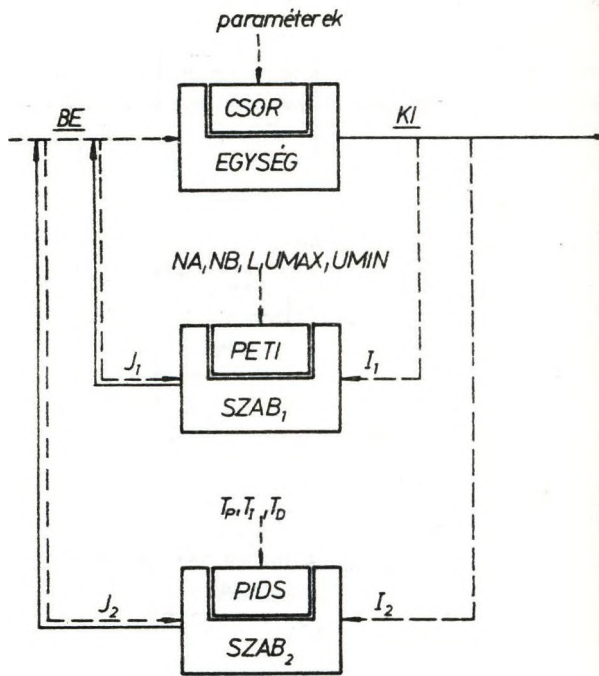
Az előadásban bemutatásra kerülő SIMULA-nyelvű szimulációs program olyan diszkrét idejű sztochasztikus szabályozott rendszerek szimulációs vizsgálatára alkalmas, amelyek egy több-bemenetű több-kimenetű szabályozott szakaszból, egybemenetű-egykimenetű szabályozókból és egybemenetű-egykimenetű identifikációs algoritmusokból, mint építőelemekből építhetők fel.

Konkrét példákon keresztül mutatjuk be, hogyan szimulálható a programrendszer segítségével különböző belső szerkezetű és egymással különböző kapcsolatban lévő diszkrét idejű sztochasztikus szabályozott rendszerek viselkedése.



3. ábra

Egy identifikáló elemmel kibővített
szabályozott rendszer



4. ábra

2 bemenetű - 2 kimenetű rendszer szabályozása

Irodalomjegyzék

- [1] Åström, K.: Introduction to stochastic control theory.
Academic Press, New York, 1970.
- [2] Csáki F.: Fejezetek a szabályozástechnikából. Állapotegyen-
letek.
Műszaki Könyvkiadó, Budapest, 1973.
- [3] Czulek A.: Szakaszos és folyamatos működésű elemeket tartal-
mazó bonyolult rendszer modellezése
Egyetemi doktori értekezés, Budapest, 1978.
- [4] Czulek A., Hangos K.: Bonyolult folytonos és félfolytonos
technológiai folyamatok modellezése.
MTA SzTAKI Zárójelentés, Budapest, 1978.
- [5] Sternby, J.: A regulator for time-varying stochastic systems;
7th Triennial World Congress of IFAC, Helsinki, 1978.
- [6] Papaverin üzem számítógépes folyamatirányítási mintarendszer
kidolgozása. A folyamatirányítási mintarendszer nagyvonalu
terve.
MTA SzTAKI - CHINOIN Jelentés, Budapest, 1975.
- [7] Hangos K.: Egy diszkrét, lineáris önhangoló szabályozó al-
goritmus szimulációs vizsgálata.
MTA SzTAKI Jelentés, Budapest, 1978.
- [8] Bányász Cs., Kéviczky L.: Discrete-time identification of
linear dynamic processes
MTA SzTAKI Tanulmányok 84/1978.

Szakaszos és folyamatos elemeket tartalmazó bonyolult
vegyipari rendszerek szimulációja

Czulek Attila^{*}

Bevezetés

A vegyipari termelés mintegy háromnegyed része szakaszos és félfolyamatos üzemekben folyik [1]. Ennek ellenére a vegyipari rendszerek vizsgálatával foglalkozó irodalom túlnyomó többsége folyamatos technológiákra vonatkozik [2].

A folyamatos technológiák térben strukturált rendszerek. Szimulációs vizsgálatukra számos flowsheeting programrendszert dolgoztak ki /pl. [3]/. Ezek azonban az eljárásmodok közötti különbségek következtében nem alkalmazhatók szakaszos technológiai folyamatok szimulációs vizsgálatára.

A szakaszos technológiák térben és időben strukturált rendszerek. Időbeli strukturáltságuk egy újabb szabadsági fokot jelent, amelyet egyre inkább szeretnénk kihasználni. Bonyolultságuk és áttekinthetetlenségük miatt az ilyen típusú rendszerek időbeli működésének vizsgálata számítógépes szimulációt igényel.

Előadásunkban szakaszos és folyamatos működésű elemeket /alrendszereket/ egyaránt tartalmazó vegyipari rendszerek modellezésével és szimulációs vizsgálatával foglalkozunk.

1. Szakaszos és folyamatos működésű elemeket tartalmazó
vegyipari rendszerek

Egy vegyipari technológiai folyamatot olyan rendszernek tekintünk, amelynek elemei a berendezések, ezek együttes, összehangolt működését az őket összekötő anyagi és in-

^{*}MTA Számítástechnikai és Automatizálási Kutató Intézet,
Budapest.

formációs csatornák biztosítják [4].

Működési sajátosságaik alapján a rendszer elemei két csoportba sorolhatók [5]:

A folyamatos működésű elemek olyan berendezések, amelyekben az anyag megszakítás nélküli áramban halad keresztül. Ennek következtében az átalakítást jellemző tulajdonság - stacioner esetben - csak a helynek függvénye, de nem függvénye az időnek. A folyamatos működésű elemek többnyire egyedi célberendezések.

A szakaszos működésű elemek olyan berendezések, amelyekbe jól meghatározott anyagmennyiséget adagolnak és az a berendezésben marad a kívánt mértékű átalakítás megtörténtéig. Az átalakítást jellemző tulajdonság az elemen belül nem függvénye a helynek, de függvénye az időnek. A szakaszos működésű elemek nagyrészt tipizált, több célra felhasználható univerzális készülékek.

A technológiai folyamatban betöltött szerepük szerint két típusukat különböztetjük meg:

A ciklikus működésű szakaszos elemek működésének egy ciklusa a betáplálást, az anyagátalakítást és az elvételt foglalja magában. Az elem működésének megkezdésére akkor kerülhet sor, ha fogadóképes és a bemenetén kapcsolódó elemek rendelkezésére bocsátják a szükséges anyagáramokat. A működés befejezésének szükséges feltétele a kívánt mértékű átalakítás megtörténte, elégséges feltétele pedig az, hogy a kimenetén kapcsolódó elemek képesek legyenek fogadni a kilépő anyagáramait. Amennyiben a betáplálási, ill. elvételi feltételek nem állnak fenn, az elemnek várakoznia kell.

A nem ciklikus működésű szakaszos elemek olyan közbülső tárolók, amelyek a ciklikus elemek kapacitás és műveleti idő különbségeiből adódó eltérések kiegyenlítésére, esetenként a minőségi ingadozások csökkentésére szolgálnak.

Lényegében passzív kiszolgáló elemek, működési készségeket saját állapotuk határozza meg.

Konkrét technológiai folyamatokon belül az elemek közötti kapcsolatok rendkívül sokfélék lehetnek. Térbeli struktúrájukat illetően a rendszerek sorbakapcsolt elemekből álló műveleti egységsorokból, párhuzamosan kapcsolt műveleti egységsorokból, közös elemeket tartalmazó párhuzamos műveleti egységsorokból és recirkulációs körökből épülnek fel [6]. A szakaszos technológiai folyamatok térbeli strukturáltságuk mellett időben is strukturáltak. Az elemek közötti kapcsolatok időben nem állandók, a működési feltételeknek megfelelően változnak. Az ilyen típusu rendszerek működése során a rendszer elemeiben nem azonos ideig tartó párhuzamos folyamatok játszódnak le. A szakaszos elemek működésük megkezdésekor, majd befejezésekor lépnek kapcsolatba a környezetükkel. Az elemek együttműködését jelentő betáplálási és elvételi folyamatokat az elem input-output működésének, az elem belsejében lejátszódó folyamatokat az elem lokális működésének nevezzük.

Az elemek működésének megkezdése feltételektől függ, működésük befejeztével új feltételek állnak elő. A rendszer elemeinek együttes, összehangolt működését szabályozó feltételek dinamikusan, eseményekhez kötött diszkrét időpontokban változnak meg.

Az összefonódó működés következtében az egyes elemek modelljeinek számítási sorrendjét nem lehet közvetlen módon előre megadni, a számítás működési feltételek szerinti automatikus vezérlését a szimulációs modellben kell megoldani.

2. A szakaszos működésű elemek modellezése

A rendszer két szakaszos működésű eleme között akkor jöhet létre anyagáramlás, ha az elemek kapcsolódási pontján a betáplálásra és az elvételre vonatkozó feltételek együttesen fennállnak. Az i -edik elemből a j -edik elembe történő be-

táplálás /elvétel/ feltételi egyenlete az alábbi logikai függvény

$$z_{i,j} = f_1(k_i(x_i), b_j(x_j)), \quad /1/$$

ahol

$z_{i,j}$ - az i -edik elemből a j -edik elembe menő anyagáramra vonatkozó megengedés, logikai változó. Értéke 1 /igaz/, ha a feltételek fennállnak és 0 /hamis/ ellenkező esetben. $z_{i,j} \equiv 0$, ha az i -edik elemből nem vezet áramot a j -edik elembe,

k_i - az i -edik elem elvételjelző logikai változója,

b_j - a j -edik elem betáplálásjelző logikai változója,

x_i, x_j - az i -edik és j -edik elem állapotváltozó vektora,

$i \in I, j \in I$ és $i \neq j$, ahol I az elemek halmaza.

Az anyagáramlásra vonatkozó feltételi egyenletek felhasználásával a j -edik elem lokális működését az alábbi állapot-egyenletek írják le:

$$\dot{x}_j = f_2(x_j, u_j(z_{i,j})), \quad /2/$$

ahol a bemenő változók u_j vektora:

$$u_j = \begin{cases} (u_j^{/1/}, u_j^{/2/})^T; & \text{ha } z_{i,j} = 1 \\ (0, u_j^{/2/})^T; & \text{ha } z_{i,j} = 0 \end{cases}$$

$i \in B_j$, ahol B_j a j -edik elemhez bemeneten kapcsolódó elemek halmaza.

Az összefüggésben szereplő $u_j^{/1/}$ a belépő anyagáramokkal kapcsolatos bemenő változó vektora, $u_j^{/2/}$ a technológiai paraméterek vektora.

$$y_j = f_j(x_j, (z_{j,k})), \quad /3/$$

ahol a kimenő változók y_j vektora:

$$y_j = \begin{cases} y_j; & \text{ha } z_{j,k} = 1 \\ \underline{0}; & \text{ha } z_{j,k} = 0 \end{cases}$$

$k_j \in K_j$, ahol K_j a j -edik elemhez kimeneten kapcsolódó elemek halmaza.

Az előbbieket szerint a j -edik elem formális matematikai modelljét a $B_j \cup K_j$ számosságának megfelelő számú /1/ feltételi egyenletek, valamint a /2/ és /3/ állapotegyenletek együttesen alkotják. Látható, hogy a betáplálási és elvételi feltételek együttes fennállása esetén ($z_{i,j} = z_{j,k} = 1; \forall i$ és $\forall k$) a szakaszos elem modellje a folyamatos elem jólismert modelljébe megy át.

A rendszermodell vezérlésének alapjául az /1/ típusú feltételi egyenletek szolgálnak. Az elemek modelljei közötti vezérlésátadást lebonyolító vezérlőrendszer kétféleképpen is kialakítható. A rendszermodell egészére vonatkozó vezérlőrendszer a vizsgált rendszer strukturájától függ, mert abban az értelemben, hogy bármilyen változtatás esetén érvényét veszti. Az előbbinél rugalmasabb megoldást jelent az egyes elemek modelljeihez rendelt, a kapcsolódó elemek közötti vezérlésátadást lebonyolító aktiválási mechanizmus kialakítása. Utóbbit az teszi lehetővé, hogy az adott elemhez tartozó feltételi egyenletekben csak az adott elem és a hozzá kapcsolódó elemek állapotától függő változók szerepelnek. Az aktiválási mechanizmus a kapcsolódó elemek között kétirányú és kölcsönös vezérlésátadást tesz lehetővé, ezáltal a kapcsolódó elemek láncolatán át biztosítja a rendszermodell vezérlését a működés folyamán kialakuló feltételeknek megfelelően.

3. Szakaszos működésű elemek közötti folyamatos működésű elemek modellezése

Folyamatos működésű elem /vagy folyamatos elemekből álló alrendszer/ szakaszos elemekhez illesztésének feltétele az, hogy az elem /alrendszer/ be-, illetve kimenetén biztosított legyen a folyamatos anyagáramlás. Ez párhuzamosan kapcsolt szakaszos elemek megfelelő működtetésével elérhető [7].

Az illesztést a gyakorlatban általában úgy valósítják meg, hogy a folyamatos elem bemenetéhez, illetve kimenetéhez két vagy több adagoló-, ill. gyűjtőtartályt kapcsolnak, amelyeket felváltva működtetnek. Az egyik adagolótartályról a másikra történő átkapcsoláskor a folyamatos elembe belépő összetétel ugrásszerűen változik meg, és - ha az adagoló tartályban nem alakul ki koncentráció-gradiens - állandó értékű marad a következő átkapcsolásig, miközben a kilépő összetétel folytonos függvény szerint változik.

A transzportegyenletekből kiindulva elkészíthető a folyamatos működésű elem diszkrét lokális működési modellje, amely az egyes adagoló tartályokról történő betáplálások függvényében írja le az elem működését [8].

A folyamatos működésű elem az éppen működő adagoló- és gyűjtőtartállyal együtt szakaszos működésű alrendszert alkot, amelynek modellje a ciklikus működésű szakaszos elem modelljére vezethető vissza.

4. A rendszermodell moduláris felépítése

Az elemek modelljei két részre az input-output működést és a lokális működést leíró részmodellekre bonthatók. Az elemek modelljeihez rendelt aktiválási mechanizmus révén az input-output modellek hálózatfüggetlenek. Az egyes részmodellek moduláris megfogalmazása nagymértékben leegyszerűsíti a modell-építkezést. A felhasználó számára

lehetővé teszi, hogy a modulok birtokában különösebb programozási ismeretek nélkül, de az építkezés szabályait ismerve, hozzá létre a vizsgálni kívánt rendszer modelljét.

A hálózatfüggetlenség és a moduláris építkezés elvén alapuló szimulációs módszer gépi megfogalmazásához a SIMULA 67 általános célú szimulációs programnyelvet [9] választottuk. A SIMULA 67 rendszerszemléletű nyelv, amelynek számos előnyös tulajdonsága közül kettőt, a párhuzamos folyamatok kvázi-parallel szimulációját biztosító időlista kezelését, valamint a moduláris építkezést nagymértékben megkönnyítő osztályszerkezetét használtuk ki.

A részmodellek funkcióinak megfelelően két alapmodult definiáltunk:

Az input-output működési modul az elem saját működésének szervezését, a kapcsolódó elemek modelljeinek aktiválását, a betáplálásokkal és elvétellel kapcsolatos akciók lebonyolítását és a lokális működési modul megfelelő időpontban történő behívását végző algoritmusokat tartalmazza.

A lokális működési modul az elem belsejében lejátszódó folyamat modelljének algoritmusát tartalmazza.

A részmodellek az általános tulajdonságok mellett speciális tulajdonságaikban különböznek egymástól. Pl. az elemek input-output működése eltér a kapcsolódások jellegét, az anyagáramok betáplálási- és elvételi stratégiáját illetően. Ez a modellben a feltételi egyenletek megoldási algoritmusainak különbözőségét vonja maga után. Ezen különbségeknek megfelelően további részmodulok hozhatók létre.

A modulokat a nyelv osztályaiban és részosztályaiban irtuk le. Mivel a részosztályok öröklik az osztály tulajdonságait, a modulok fastrukturájú hierarchia-rendszert alkotnak. Ezek birtokában a vizsgálni kívánt rendszer modellje

modulárisan építhető fel, az elemek modelljei a tulajdonságokat tartalmazó modulokból, a rendszer modellje a hálózatfüggetlen moduláris elemekből.

A rendszermodell felépítésének ez a módja lehetővé teszi a modell strukturájának egyszerű bővítését, amely nem jelent mást, mint az új elem modelljének generálását és illesztését a már meglévő rendszermodellhez. Erre akár a szimuláció végrehajtása közben is lehetőség van.

5. Alkalmazás

A vegyipar alapvetően szakaszos gyártásra berendezkedett ágazataiban /gyógyszer-, lakk-festék-, műanyagipar, stb./ számos olyan technológiai folyamatot ismerünk, amelyek a szakaszos elemek mellett folyamatos működésű elemeket is tartalmaznak. Az ismertetett módszer ilyen típusú rendszerek tervezési, folyamatfejlesztési és irányítási célú, rendszeranalízis jellegű szimulációs vizsgálatainak elvégzésére alkalmas. Alkalmazása elsősorban olyan esetekben célszerű, amikor a rendszer bonyolultsága meghaladja az emberi áttekinthetőséget.

Tervezés

Általános tapasztalat, hogy a szakaszos technológiai folyamatok tervezésekor egyes berendezéseket - főként a közbülső tároló tartályokat - alul- vagy túlméreteznek. Az üzemelés szempontjából nagy problémát okoz ez akkor, ha a rendszerben több recirkuláció is van. A tervezett rendszer időbeli működésének szimulációja ilyen esetekben lehetővé teszi az optimális készülékméret megválasztását.

Félüzemi és kísérleti üzemi mérések üzemelési tapasztalatok birtokában lehetőséget biztosít a különböző tervezési változatok várható működésének vizsgálatára és az összehasonlítás alapján az optimális változat kiválasztására.

Folyamatfejlesztés

A jelenlegi igényeket és lehetőségeket figyelembe véve a folyamatfejlesztés az a terület, ahol a módszer alkalmazásával nyert eredmények a leggyorsabban realizálhatók.

Termelő üzem esetében a szimulációval a működés megzavarása nélkül feltárhatók a rendszer szűk keresztmetszetei [10]. A kapacitásbővítés lehetőségeinek megfelelően a rendszermodell új berendezések modelljeivel bővíthető. Az egyes alternatívák szerint bővített rendszermodellek segítségével elvégzett szimulációs vizsgálatok összehasonlítása alapján meghatározható a rendszer bővítésének legcélszerűbb módja.

A rendszer részfolyamatainak intenzifikálása eredményeként megváltozik a rendszer időbeli működése. A szimulációval vizsgálhatók az intenzifikálás következtében a rendszer működésében várhatóan előálló változások és ennek alapján meghatározhatók a szükséges technológiai módosítások.

A szakaszos és folyamatos működésű elemekből álló rendszerek esetében nagyon gyakoriak az alap- és segédanyagok hatékony felhasználása érdekében alkalmazott recirkulációk. A szimulációs módszer alkalmazásával lehetőség van az egymással konkurráló recirkulációs stratégiák és üzemmódok összehasonlítására, optimalizálására a rendszer működésének megzavarása nélkül.

Irányítás

A közelmúltban - a viszonylag olcsó mikroprocesszoros folyamatirányító berendezések megjelenését követően - megnőtt a szakaszos technológiai folyamatok számítástechnikai eszközökkel történő irányítása iránti érdeklődés. Egy több berendezésből álló bonyolult működésű üzem folyamatirányítása kétszintű hierarchikus irányítást igényel, amely részben vezérlési, részben szabályozási funkciókat foglal magában. Ilyen esetekben a módszer alkalmazása hatékony

segítséget nyújt mind a rendszeranalízis, mind pedig a szintézis munkafázisaiban.

Irodalomjegyzék

- [1] Bakos M.: Műszaki Kémiai Rendszerek '78 Konferencia, Keszthely, 1978. Plenáris előadás.
- [2] Hlaváček, V.: Computers and Chemical Engineering, 1, 75-100 /1977/.
- [3] Benedek Pál és szerzőtársai: Bonyolult műveleti egységek matematikai szimulációja. Akadémiai Kiadó, Budapest, 1973.
- [4] Kafarov, V.V. - Perov, V.L. - Mesalkin, V.P.: Vegyipari rendszerek matematikai modellezése. Műszaki Könyvkiadó, Budapest, 1977.
- [5] László A.- Benedek P.: A vegyészmérnöki tudomány alapjai. Műszaki Könyvkiadó, Budapest, 1964.
- [6] Brodman, M.T. - Smith, C.L.: Chemical Engineering, 83, 191-198 /1976/.
- [7] Veress G. - Czulek A.: Hungarian Journal of Industrial Chemistry, 4, Suppl. 1-2, 149-154 /1976/.
- [8] Almásy G. - Czulek A. - Pallai I. - Veress G.: 7th European Symposium Computer Application in Process Development. Erlangen, Preprints 122-129 /1974/.
- [9] SIMULA 67 Common Base Language Norwegian Computing Centre, Oslo, 1968.
- [10] Czulek A. - Ser V. - Knuth E.: Mérés és Automatika, 26, 3, 95-97 /1978/.
- [11] Czulek A.: Szakaszos és folyamatos működésű elemeket tartalmazó bonyolult rendszer modellezése. Műszaki doktori értekezés, Budapest, 1978.

Folytonos-diszkrét kombinált szimulációs rendszer-vizsgálatok

Csáki Péter

B e v e z e t é s

A számítógépes szimuláció a dinamikus rendszerek vizsgálatának sok területen bevált módszere. A szimuláció segítségével a rendszer helyett annak számítógépes modelljén végezzük a vizsgálatokat és ilyen módon gyorsabban és kisebb ráfordítással szerezhethünk információkat a rendszer tulajdonságairól.

Egy adott dinamikus rendszer szimulációját a rendszer modellje alapján végezzük. Ezért mindenekelőtt a rendszer modelljét kell kidolgozni. A modellnek tartalmaznia kell a rendszer struktúrájának és a struktúrális összefüggéseknek, valamint a rendszer állapotainak és az állapotváltozásokat meghatározó összefüggéseknek a leírását.

Az állapotváltozások alapján a modellek két fő típusát lehet megkülönböztetni: az állapotokat diszkrét időpontokban, illetve folytonosan változtató modellek. Pl. tömegkiszolgáló rendszerek lényeges állapotainak az erőforrások foglaltsági állapotait, a felhasználóknak a kiszolgálásukhoz viszonyított állapotait (várakozás, adott erőforrás használata stb.) tekintjük és az ezen állapotok közötti átmenet diszkrét időpontokban történik ("esemény"). Folytonos állapotváltozású modelleket alkalmazunk pl. áramlási folyamatok leírására, többnyire differencia- vagy differenciálegyenletek segítségével.

Mindkét rendszer-modell-típushoz megfelelő szimulációs nyelveket dolgoztak ki. Diszkrét, eseményorientált szimuláció céljaira a legismertebb nyelvek: GPSS, SIMSCRIPT, SIMULA; folytonos folyamat-szimuláció céljaira néhány ismert nyelv: ANDISIM, DYNAMO, CSMP.

A diszkrét, ill. folytonos típusú rendszer-modellek köre azonban nem meríti ki az összes lehetőségeket; bizonyos esetekben általánosabb rendszer-vizsgálatokra van szükség, amelyekben diszkrét és folytonos állapot-átmenetek egyaránt előfordulnak. Az említett szimulációs nyelvek többsége ilyen kombinált típusú szimulációra nem volt alkalmas, ezért az utóbbi években több erőfeszítés történt egyrészt a meglévő nyelveknek a kombinálás lehetőségeivel való kiegészítésére [2], [6], másrészt ilyen célú új nyelvek létrehozására, pl. [3], [7].

A rendelkezésre álló nyelvek között ebből a szempontból helyet foglal el a SIMULA 67 [1], [5]. T.i. amíg a többi nyelv kiegészítése csak a nyelv bővítése útján lehetséges, addig a SIMULA - általános fogalomfelépítési lehetőségei (osztály, alosztály) folytán - a kombinált típusú szimuláció a meglévő nyelvi konstrukción belül megoldható. Ezen túlmenően, a nyelv lehetővé teszi, hogy a kialakítandó nyelvi fogalmak a modell megfelelő fogalmaival összhangban lehessenek.

Az elvadásás célja egy ilyen lehetőségeket nagymértékben kihasználó kombinált típusú szimulációs software eszköz (SIMULA-osztály) ismertetése. Megjegyezzük, hogy eddig két hasonló célú SIMULA-osztály készült: az egyik COMBINED SIMULATION néven [8], amely a modell fogalmainak nyelvi megfeleltetését csak részben oldja meg; a másik CADSIM néven [4], amelynél amellet, hogy ilyen megfeleltetés egyáltalán nincsen, az állapotváltozók számitási algoritmusai szinkron-hibásnak bizonyult.

1. Dinamikus rendszer struktúrája és állapotai

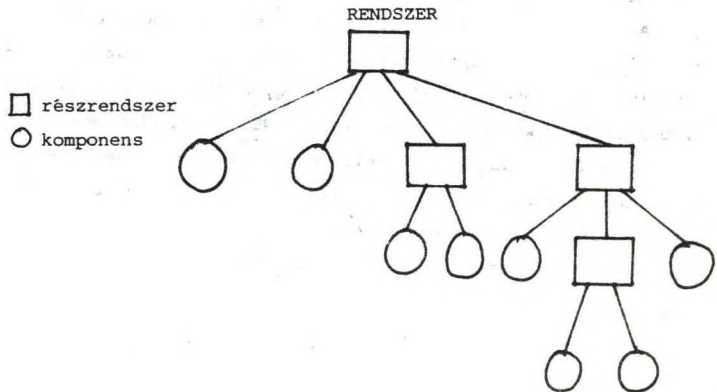
Valamely valóságos objektum modellje az objektumnak egy adott szempontból lényeges tulajdonságait emeli ki. Természetesen különböző szempontok különböző modelleket eredményeznek. A modelleknek egy fontos típusát alkotják a rendszer-modellek, amelyekben lényeges szerepe van az objektum belső struktúrájának és az objektum részei közötti kölcsönhatásoknak. Ilyenkor a modellezett objektumot rendszernek nevezzük. A rendszer időbeli viselkedését a rendszer állapotaival, illetve ezek változásával írhatjuk le.

1.1 Rendszerek struktúrája

Rendszerek vizsgálatánál a legfontosabb kérdés, hogy milyen kapcsolat van a rendszer egyes részei között. Ezért a modellben meg kell határozni a rendszer struktúrális elemeit. Ezen elemeknek két típusát különböztetjük meg:

- a.) részrendszerek, amelyek további elemekre bonthatók,
- b.) komponensek, vagy alapelemek, amelyeket már nem bontunk további elemekre.

A rendszer struktúráját a struktúrális elemeknek egy gráfjával, pontosabban egy fával ábrázoljuk:



1. ábra. Rendszer-struktúra

Az elemek egymáshoz való viszonyát a strukturális összefüggések adják meg, amelyeket különféle típusú relációkkal fejezhetünk ki: pl. tartalmazási reláció (amely a részrendszer-komponens viszonyt adja meg), környezeti reláció (amely azt adja meg, hogy mely elemek alkotják egymás közvetlen környezetét), az egyes elemeknek egymásra való közvetlen kölcsönhatásait megadó reláció, stb.

1.2 A rendszer állapotok

Minden időpontban egy adott állapot jellemző a teljes rendszerre, ennek részrendszereire és komponenseire egyaránt. Minden strukturális elemhez tartozik egy állapotter (az elem összes lehetséges állapotainak halmaza); ezek együttesen alkotják a teljes rendszer állapotterét. Amennyiben a rendszer állapota időben változik, dinamikus rendszerről beszélünk. (Számítógépes szimulációval éppen ezeket az állapotváltozásokat kívánjuk a számítógépen realizálni.) Az állapotváltozásokat vagy más szóval állapot-átmeneteket folyamatát a rendszerben érvényes dinamikai összefüggések határozzák meg, amelyek egy transzformációt írnak le az állapotterben.

A rendszer vagy egy komponensének állapotát rendszerint több tényező együttesen alakítja ki, ezeket nevezzük állapotváltozóknak. Ezek különböző típusúak lehetnek: mennyiségi, strukturális, stb. Időbeli változásuknak két alaptípusát különböztethetjük meg: időben folytonos és diszkrét időpontokban változó típusok.

2. Dinamikus rendszer szimulációja

A rendszer-szimuláció céljára szolgáló nyelvek általában olyan eszközöket nyújtanak a felhasználónak, amelyekkel megfelelően és többé-kevésbé kényelmesen algoritmizálhatja az állapot-átmeneteket megadó összefüggéseket. Kevesebb gondot fordítanak azonban arra, hogy a szimulációs programban a rendszer struktúrája is felépíthető legyen. Ennek azonban jelentős szerepe lehet több szempontból is. Egyrészt modell-egyszerűsítés vagy modell-fejlesztés esetén, amikor a modell egy részrendszerét vagy egy

egyszerű komponensét helyettesítjük egymással. Másrészt, olyan rendszerek esetén, amelyeknek a struktúrája is változik és így a struktúra átrendezése is a dinamikus leírás részévé válik, az ehhez szükséges eszközökkel rendelkező nyelven írt programba ennek algoritmusát könnyen és áttekinthetően beépíthetjük.

Az állapotátmeneteket diszkrét időpontokban illetve időben folyamatosan változtató modellek szimulációja különböző technikát igényel.

2.1 Diszkrét állapot-átmenetek

A diszkrét esemény-orientált modellek szimulációja során csak az események (állapot-átmenetek) időpontjai a lényeges időpontok. Minden komponenshez egy időpontosorozat tartozik, a komponens állapotváltozásait leíró algoritmus akciót ezen időpontokhoz rendeljük. Ezért az ilyen modellekhez kidolgozott szimulációs nyelveknek biztosítani kell, hogy az egyes komponensek a megfelelő időpontokban időzithetők legyenek, ezekben az időpontokban hajtsák végre az aktuális akciókat, valamint, hogy a komponensek az időzítés sorrendjében kapják meg a vezérlést.

2.2 Folytonos állapot-átmenetek

Folytonos folyamatok szimulációja során az előrehaladás az időben megfelelően kicsiny lépésközökkel, rendszerint valamilyen közelítő algoritmussal történik. Ilyenkor biztosítani kell a modell típusának megfelelő haladási algoritmust, ennek megfelelő

pontosságát, valamint a lépésköz megfelelő megválasztásának lehetőségét.

2.3 Diszkrét és folytonos állapot-átmenetek

A folytonos folyamatok bizonyos időpontig követett törvényszerűségei valamilyen "esemény" hatására megváltozhatnak. Ugyanakkor a folytonos folyamat valamely állapota is indukálhat egy "esemény"-t. Az ilyen modellek szimulációja során a 2.1. és 2.2. pontokban említett követelményeken túlmenően az állapotukat diszkrét időpontokban és a folyamatosan változtató komponensek ezen kölcsönhatását is figyelembe kell venni; biztosítani kell a folytonos folyamat lépésközének automatikus megváltoztatását, ha túllépné egy őr befolyásoló esemény időpontját. A problémát a következő példával világíthatjuk meg.

2.4 Kohó szimuláció

Egy kohóban acéltömböket hevitünk adott hőfokra. Az acéltömböknek a kohóban való tartózkodását hőátadási differenciálegyenletek szabják meg, amelyek a kohó és az acéltömbök hőmérsékletének változásait írják le (folytonos folyamatok). Az acéltömbök érkezését, a kohó befogadóképességét meghaladó számban érkező tömbök kohó előtti várakozását, a kohóba helyezését és távozásukat viszont diszkrét eseményekkel írhatjuk le. Ebben a modellben egy acéltömb kohóba helyezése megváltoztatja a kohó hőátadási viszonyait. Továbbá, ha egy acéltömb eléri a kívánt hőfokot, távozik a rendszerből és ugyancsak megváltoznak a hőátadási viszonyok.

3. DISCOS: a kombinált típusú szimuláció egy eszköze

Az 1. pontban kiemeltük és röviden vázoltuk a rendszerelméletnek azokat a fogalmait, amelyek a kombinált szimuláció céljára DISCOS (Discrete/Continuous Simulation) néven kidolgozott eszközben lényeges szerepet játszanak.

A DISCOS különbözőképpen kezeli a diszkrét, ill. a foly-

tonos típusú komponenseket. A diszkrét típusú komponensek önállóan időzithetők a szimulációs időtengelyen. A folytonos típusú komponensek azonban mind azonos időpontra, a mindenkori jelenben vannak időzítve. A továbbiakban ez utóbbiakkal foglalkozunk részletesen.

3.1 A folytonos komponensek típusai

Folytonosnak tekintjük egy rendszer azon komponenseit, melyek állapotait egyetlen valós változóval adhatjuk meg.

A DISCOS-ban^a folytonos komponensek lehetnek determinisztikusak, ill. sztochasztikusak. A determinisztikus komponensek állapot-átmeneteit meghatározhatják deifferencia- ill. differenciálegyenletek. Mindegyik tipushoz különböző állapotátmenet - számítási módszerek tartoznak, amelyek közül a felhasználó kiválaszthatja az adott modellhez megfelelőt, de szükség esetén új módszereket is megadhat programjában.

A felhasználó minden komponenshez ill. részrendszerhez külön adhatja meg állapot- és egyéb változókat, valamint az állapotátmeneteik egyenleteit. A DISCOS eszközt szolgálták a folytonos komponensek és részrendszerek (1.ábrán szemléltetett) tartalmazási gráfjainak felépítésére és változtatására bármely időpontban. Az állapotátmenetek számításakor ennek sorrendje is lényeges lehet, ezt a gráfban az egyes részrendszerekhez tartozó elemek sorrendje határozza meg. A sorrendet a felhasználónak módja van megadni, sőt, akár menetközben megváltoztatni.

3.2 A folytonos folyamat haladása az időtengelyen

A folytonos komponensek egyes időintervallumokban aktívak, másokban passzívak. (Utóbbi esetben csak diszkrét komponensek vannak időzítve az időtengelyen.) Folytonos folyamat csak akkor halad az időtengelyen, ha van a rendszerben aktív folytonos komponens. A felhasználó START utasítással a megfelelő időpontban aktivizálhat

egy folytonos részrendszert és ezáltal a folytonos folyamatok beindulnak. A STOP utasítás hatására pedig az aktiv folytonos részrendszer passzíválódik.

Ha a folytonos folyamat aktiv, akkor az aktiv komponensek típusainak megfelelő módszerek működnek. (Egyelőre differenciálegyenlet-rendszerekre Euler és Runge-Kutta módszerek, a sztochasztikus folyamatok közül pedig stacionárius Gauss-Markov-folyamatokra állnak rendelkezésre módszerek.) A szabad haladás lépésköze vagy fix hosszúságú, vagy a pontosság (szórás) függvényében változhat, a felhasználó által választható módon.

Ha a szabad haladás lépésközén belül olyan esemény (diszkrét komponens) van időzítve, amely megváltoztatja a folytonos folyamat állapot-átmeneti törvényeit, vagy felhasználja a folytonos állapotváltozók pillanatnyi értékeit, akkor a szóbanforgó esemény befolyásolja a folytonos folyamat haladását. Ennek a helyzetnek a kezelésre a következő lehetőségek közül választhatunk:

a) a folytonos folyamat minden lépése előtt megvizsgáljuk, hogy van-e esemény időzítve a szóbanforgó intervallumban. Ha igen, a lépésközt lerövidítjük oly módon, hogy a folytonos folyamat ezen esemény időpontjára lépjen;

b) a szóbanforgó diszkrét komponens figyel, hogy a folytonos folyamat túllépte-e őt. Ha igen, akkor a folytonos folyamat megismétli a lépést, csökkentve lépésközzel;

c) ha a szóbanforgó diszkrét komponens csak felhasználja a folytonos állapotváltozók értékeit, akkor erre a célra interpolációt végez;

d) "figyelő"-ket időzítünk az időtengelyre az esemény időpontja előtt "lépésköz" távolsággal, amelyek gondoskodnak a lépésköz csökkentéséről.

A felsorolt lehetőségek egyike sem tekinthető optimálisnak a többivel szemben. Az a megoldás előnye, hogy teljesen automatizálható, ezzel szemben hátránya, hogy a vizsgálatot minden lépésben el kell végezni, függetlenül ennek eredményétől. A többi

megoldásnál a felhasználónak kell gondoskodnia a vizsgálat elvégzéséről, de csak olyankor, amikor ^{ésszerűen} szükséges. A d) megoldás előnye a b)-vel szemben, hogy nincs szükség a lépés megismétlésére, hátránya viszont, hogy több eseményhez tartozó egyidejűleg aktív "figyelő" különböző nehézségeket okoz a lépésköz megfelelő csökkentésének megvalósításában. A c) megoldás is kiküszöböli az ismétlést, de sok érték interpolálása esetén több gépidőt igényel. Ezért az egyes megoldások csak ezek további tesztelése után értékelhetők.

4. A DISCOS beágyazása a SIMULA nyelvbe

A DISCOS legcélszerűbben a SIMULA nyelvre és annak SIMULATION osztályára alapozható, mivel ezek szolgáltatják a legtöbb olyan eszközt, amellyel a DISCOS az ismertetett tulajdonságaival felépíthető.

Ilymódon a DISCOS egy SIMULA-osztály, pontosabban a SIMULATION alosztálya:

```
SIMULATION CLASS DISCOS;BEGIN .....END;
```

A diszkrét komponensek a DISCOS-on belül is PROCESS-alosztályú objektumok, amelyek használatában semmilyen eltérés nincs. A folytonos folyamatot egyetlen PROCESS-alosztályú objektum képviseli CONTINUOUS azonosítóval.

A folytonos részrendszereket és komponenseket LINK alosztályú objektumokkal adhatjuk meg:

```
LINK CLASS CONTINUOUS ELEMENT .....;
CONTINUOUS ELEMENT CLASS COMPONENT .....;
CONTINUOUS ELEMENT CLASS SUBSYSTEM .....;
```

A SUBSYSTEM osztályhoz tartozik egy HEAD osztályú objektum, ilymódon a rendszer struktúrája (tartalmazási gráfja) SIMSET-listák segítségével építhető fel.

Az állapot-átmenet-számítási módszerek szintén osztályokba vannak foglalva:

HEAD CLASS CONTINUOUS METHOD;

CONTINUOUS METHOD CLASS INTEGRATION METHOD;

és így tovább, minden módszerhez. A

CLASS ACTUAL METHODS;

osztály biztosítja a pillanatnyilag aktuális módszerek kombinációjának nyilvántartását és különböző eljárások lehetővé teszi, hogy a felhasználó - ha szükségesnek látja - valamely időpontban áttérjen másik módszerre.

A DISCOS osztály tartalmazza a

REAL STEP, MINSTEP, MAXSTEP

globális valós attribútumokat, ahol STEP a pillanatnyi lépésköz, MINSTEP és MAXSTEP a felhasználó által megadott legkisebb, ill. legnagyobb lehetséges lépésköz. Ezekhez a felhasználó közvetlenül nem férhet hozzá (HIDDEN), csak megfelelő eljárásokon keresztül.

Ez a rövid vázlat csak a DISCOS leglényegesebb tulajdonságait sorolja fel; részletesebb ismertetésre itt nem térhetünk ki.

5. A DISCOS összehasonlítása más SIMULA-beágyazásokkal

5.1 CADSIM

A CADSIM nevű SIMULA-osztály leírása [4] -ben található. (Az országban az MTA CDC 3300 gépén hozzáférhető.) Legfontosabb alkotórésze:

PROCESS CLASS CONTINUOUS(N);;

osztálydeklaráció. Ennek segítségével adhatók meg (differenciálegyenletek formájában) a folytonos folyamat alkotóelemei. Tartalmaz egy INTEGRATE nevű eljárást, amely a folytonos folyamat haladását biztosítja az időtengelyen.

Ez a felépítés nem szerencsés több szempontból:

- a) Az N paraméterrel kell megadni az egyenletek számát, ami menetközben már nem változtatható.
- b) Több CONTINUOUS-objektum haladhat az időtengelyen, ami lehetővé teszi, hogy ezek különböző időpontokra legyenek időzítve és ezáltal állapotváltozók pillanatnyi értékei más időpontra érvényesek. Ilyenkor a számított értékek hibásak.
- c) Az INTEGRATE eljárást a felhasználó nem megfelelő környezetben is hívhatja, ezáltal felborulhat a PROCESS-objektumok vezérlés-átadási rendje (hibás programfutást eredményezve anélkül, hogy a felhasználó erről hibajelzést kapna).

5.2 COMBINEDSIMULATION

Felépítése különbözik a CADSIM-től. A folytonos folyamat képviselőjében egy PROCESS-alosztályú "monitor" halad az időtengelyen. (Ez megfelel a DISCOS-ban a CONTINUOUS objektumnak, de funkciója részben különbözik attól).

Minden állapotváltozóhoz egy

LINK CLASS VARIABLE (STATE); REAL STATE

osztályú objektumot rendel, amelyet (a megfelelő időpontban) START eljárással lehet aktivizálni. A dinamikai egyenletek a

LINK CLASS CONTINUOUS;

osztály alosztályaiban adhatók meg (a megfelelő VARIABLE-objektumokra hivatkozva). Ezen objektumok szintén START eljárással aktivizálhatók és STOP eljárással passzivizálhatók.

Ennek a felépítésnek a hátránya egyrészt, hogy "részrendszerek" nem képezhetők, csak "komponensek", másrészt a komponensek attribútumait és dinamikai egyenleteit különválasztva kell megadni.

A folytonos folyamat és a diszkrét komponensek kapcsolatát mindkét rendszer a 3.2.-ben említett a) változattal oldja meg. (Tehát a folytonos folyamatot minden esemény megállíthatja.)

I R O D A L O M

- [1] G.M.Birtwistle et al: SIMULA Begin. Auerbach 1973.
- [2] F.E.Cellier, A.E.Blitz: GASP-V: A Universal Simulation Package. In: L.Rekker:Simulation of Systems.North Holland,1976.391-402.
- [3] F.E.Cellier: COSY- a combined system simulation language, 1977.
- [4] R.J.Cunningham, F.N.Parr, R.J.W.Sim: An introduction to combined continuous system and discrete event process simulation in SIMULA. Imperial College Publ.76/11.1976.
- [5] O.J.Dahl, B.Myhrhang, K.Nygaard: SIMULA-67 Common Base Language.NCCPubl.S-22.1971.
- [6] M.C.Delfosse: Continuous Simulation and Combined Simulation in SIMSCRIPT II.5. C.A.C.I.Publ.1976.
- [7] D.G.Golden, J.D.Schoeffler: GSL-A combined continuous and discrete simulation language. SIMULATION 1973.1-8.
- [8] K.Helsgaun: COMBINEDSIMULATION. 6-th SIMULA Users' Conf. Lisboa, 1978.

Diszkrét vezérlési rendszerek modellezése

Knuth Előd, Sárközy András *

1. Bevezetés

A dolgozat rövid ismertetést ad egy SIMULA nyelvre alapított általános célú diszkrét rendszerelemzési eszközzől. Ez egy hierarchikusan felépített fogalmi könyvtárból, a könyvtár fogalmaira épített célnyelvből, és az ezek kezeléséhez szükséges software eszköztárból áll.

A kidolgozott módszer segítségével diszkrét kommunikáló folyamatrendszerek vizsgálhatók. A kommunikáció u.n. "message buffer"-ek segítségével valósul meg, melyek egyúttal a folyamatok közötti szinkronizáció céljait is szolgálják.

A vizsgálható tulajdonságok lehetnek logikai és statisztikai természetűek. Ezek egy eléggé széles standard választékának alkalmazását automatikus eszközök segítik. Az automatizálás kiterjed a nyert eredmények dokumentálására is.

2. Fogalmi szintek

A rendszer alapját képező fogalmi könyvtár három egymásra épített szintből áll. Az első szinten csak igen általános alapfogalmak szerepelnek, és csupán olyan tulajdonságaik formalizálásával, melyek minden konkrét alkalmazás szempontjából közösnek tekinthetők. A fontosabb leírt fogalmak a

* MTA Számítástechnikai és Automatizálási Kutató Intézet

következők: véges folyamat, ciklikus folyamat, passzív ciklus, időzített ciklus, vezérelt ciklus, monitor, adminisztrációs objektum. E fogalmak úgy vannak megadva, hogy csupán sémájuk rögzített, így konkrét tartalmuk a hierarchia későbbi szintjein adható meg.

A második szinten a leggyakrabban használt adminisztrációs típusokat vezettünk be. Ezek osztályozzák, és egy külön hierarchikus rendszerbe foglalják a statisztikai megfigyelések tipikus módszereit.

A harmadik szinten specifikáltunk egy lehetséges kommunikációs és szinkronizációs eszköztárat. Ezek egyuttal olyan adminisztrációs objektumokkal vannak automatikusan ellátva, melyek a lezajlott vezérlési akciókról nyilvántartást készítenek. Ilyen módon minden kommunikációs ponton választ kapunk a torlódások, várakozások mértékéről, továbbá a működés logikai korrektségéről.

3. Vezérlési séma

A harmadik szinten megadott eszközök az alábbi sémába illeszkednek: üzenettárolók egy tetszőleges megadott rendszerét tekintjük. E tárolók bármelyikébe bármely folyamat üzenetet helyezhet el, /esetleg többet is egyidejűleg, különböző vagy akár azonos tárolókba is/. Az üzenetek azonosítókkal vannak ellátva, ami lehetővé teszi, hogy a küldő folyamatok bizonyos akcióit az egyes üzenetek sorsától tegyék függővé. Ennek legtipikusabb esete a feltételes felfüggesztés.

Megjegyezzük, hogy vannak olyan alkalmazások, melyekben utólag nem azonosítható üzenetekkel is kielégítően lehet dolgozni. Ennek eszköztára az általunk megadottnál lényegesen egyszerűbb is lehet.

Az üzenettárolók tartalmának feldolgozása más folyamatok hatáskörébe tartozik. Ezek gondoskodnak arról is, hogy az elküldött üzenetjük feldolgozására váró felfüggesztett folyamatok ismét meginduljanak.

4. Modell szerkezet

A leírandó rendszerek alkotó elemeit célszerű a logikailag összetartozás alapján önálló, nagyobb egységekbe foglalni. Ennek az alábbiakban ajánlott módjához eszközöket is biztosítottunk.

Rendszer komponensnek tekintünk egy olyan nagyobb egységet, mely egy vezérlő folyamatból /monitor/, ennek alárendelt további folyamatokból, továbbá a hozzájuk tartozó adminisztrációs és kommunikációs objektumok halmazából áll.

Ilyen komponensnek tekinthető például egy operációs rendszer a magját képező monitorral, funkcionális rutinjaival és egyéb tartozékaival együtt. Ilyennek tekinthetünk azonban egy egyszerű születés-kihalási folyamatot is, a folyamatgenerátorral, magukkal a generált folyamatokkal, és megfelelő statisztikai megfigyelésekkel együtt.

5. Gépi realizáció

Az előző pontokban vázolt modell elemzési módszer az MTA központi CDC 3300-as gépén SIMULA 67 nyelven megvalósításra került. Ennek vázlatos felépítése a következő:

- a/ Rendelkezésre áll egy fogalmi könyvtár, melyet egy cél-orientált modell leíró nyelv egészít ki. Az ezen a nyelven leírt modelleket egy előfordító SIMULA nyelvre fordítja.
- b/ Az így létrejött szimulációs program a specifikált megfigyelések eredményeit sűrített, kódolt formában egy közbülső tárolóra helyezi.
- c/ Működés során a program futása megszakítható, és esetleges változtatások után továbbindítható, vagy más paraméterekkel ismételhető.
- d/ A közbülső tárolóra helyezett kódolt információk feldolgozását egy standard utófeldolgozó végzi, mely fel van készítve a standard megfigyeléstípusok mindegyikére. Az eredményeket statisztikailag kiértékeli és dokumentálja.

