

E. van Elderen

PERSONAL PROSIM is een simulatie taal voor continue en discrete veranderingen in een systeem; de keuze voor deze gecombineerde taal voor DLO-instituten is gemaakt door de DLO-projectgroep "Evaluatie Simulatie Software". De taal is sterk gestructureerd in objecten (class componenten) en hanteert de proces-beschrijvingsmethode per object om de voortgang in de tijd te beschrijven. Het bouwen van een model gebeurt op een PC in het 'Maintenance System', dat tevens de datafiles, de simulation-environments en de import van modules van andere modellen beheert. Daarnaast is er nog een 'Run Control System' dat op verzoek een trace van gebeurtenissen levert, grafieken met variabelen presenteert of de toestand van variabelen op het scherm vertoont of verandert. Tenslotte zijn er nog de 'Supplements' voor: 'Animation' om het dynamisch gedrag te demonstreren; 'Graphics' om figuren te maken van variabelen van verscheidene runs of tegen elkaar; 'Statistics' om variabelen te transformeren, kenmerken te berekenen (gem., standaard afwijking, auto correlatie enz.), te vergelijken met standaardverdelingen of om variabelen met elkaar te vergelijken of een regressiemodel te berekenen.

Inleiding

Onderzoek gaat in op de vraag: hoe ziet de werkelijkheid eruit, hoe zit deze in elkaar? Door nauwkeurig toezien kan het gedrag van een systeem (een deel van de werkelijkheid) worden achterhaald en beschreven.

Nu is er een tendens om niet alleen te willen weten, maar om het weten ook ten nutte te maken in het beleid, d.w.z. de werkelijkheid zo beïnvloeden dat er een gewenst resultaat ontstaat. Een experiment in de werkelijkheid is echter niet altijd mogelijk. Dit is de reden om te zoeken naar modellen die bijvoorbeeld in wiskundige of logische vorm worden weergegeven. Het werken met deze modellen op een computer wordt veelal aangeduid met simulatie.

Simulatie

De definitie van simulatie is niet eenduidig, die van C.T. de Wit luidt: "*.. simulatie is het bouwen van een model en het bestuderen van zijn gedrag in vergelijking met die van het systeem.*"

Wat in deze definitie ontbreekt is het gebruik van modellen met het doel om strategieën van beïnvloeding te evalueren en om voor nog niet bestaande situaties het gedrag te leren kennen.

Essentieel voor de moderne simulatietalen is het object georiënteerd programmeren. Er wordt een 'class' gedefinieerd en van zo'n class kunnen kopieën, componenten worden gemaakt, bijvoorbeeld klanten in een winkel. Een tweede aspect is de procesbeschrijving in een 'class', waarmee de veranderingen in de tijd van elke component

in het systeem beschreven worden. Dit vergt een goede coördinatie van de afzonderlijke componenten in het simulatiemodel, omdat iedere component op een ander punt van het proces kan zijn. Omdat een gangbare computer statements één voor één afwerkt is het nodig om processen die in feite parallel verlopen in de computer quasi-parallel af te werken; dit wordt bereikt door processen tijdelijk te onderbreken en op andere momenten (in de simulatietijd) de draad weer op te nemen. Processen worden onderbroken door het toevoegen van speciale statements als: Hold t en Passivate; met Hold t wordt het betreffende proces t tijds-eenheden onderbroken en op het moment (nu + t) van de tijdas geplaatst; met Passivate breekt het proces de afwikkeling van de statements af, zonder dat er iets wordt gezegd over de hervatting daarvan.

PERSONAL PROSIM als simulatie gereedschap

De bespreking van enkele facetten van PROSIM (Release 2.02) en het werken ermee wordt opgedeeld in vier paragrafen:

- Modelling, waarbij het gaat over de door PROSIM opgelegde structuur van het model en over de taalelementen die beschikbaar zijn;
- Maintenance system, waarbij de aandacht gericht wordt op het gestructureerd omgaan met een model en de bijbehorende data files (input, output), op het gebruik van modules van andere modellen en procedures in andere talen en op het gebruik van een model op een andere PC;
- Run control system; aan de orde komen de faciliteiten van PROSIM om het proces op het scherm te volgen, de attributes (parameters, variabelen) van componenten te wijzigen en data met de omgeving uit te wisselen;
- Supplements; voor de uitwerking van resultaten van verscheidene simulatie runs zijn beschikbaar de Animation, de Graphics en de Statistics supplementen.

Modelling

De structuur van een PROSIM model bestaat uit afzonderlijke modules. Twee modules zijn voorgeschreven; het eerste, DEFINE, bevat de opbouw van het model in componenten en attributen d.w.z. de definitie sectie (declaraties) komt in één module voor; het tweede, MAINMOD, bevat de beschrijving van het gedrag van componenten en in ieder geval van de component MAIN, die nodig is om de simulatie te starten. De dynamische sectie kan volledig in de MAINMOD module worden opgenomen, maar kan ook over verscheidene modules worden verspreid.

Er zijn twee soorten attributes; de system-attributes worden automatisch toegekend en bijgewerkt, de user-defined attributes moeten worden gedeclareerd met een bepaald type (real, reference to set, inputstream, macro enz.).

De regeling van de activiteiten van processen in componenten is nodig omdat de computer niet alle processen

tegelijkertijd kan behandelen. De process control statements zijn bijvoorbeeld:

```
ACTIVATE C WITH DELAY 5 MINUTES
IF C IS NOT ACTIVE
C wordt dan in de event-lijst gezet op 5
minuten na simulatietijd NOW;
HOLD t;
WAIT WHILE TEMPERATURE OF BAR 500 WITH
ACCURACY 3.
```

Zo is er slechts één proces current en na het afwerken van de statements wordt in de event-lijst gekeken welk proces vervolgens current moet worden. Tegelijkertijd wordt dan zonodig de simulatietijd verhoogd.

Queues en sets worden gebruikt om componenten te verzamelen in een wachtrij of verzameling. 'ENTER Q', 'REMOVE C FROM Q' en 'Q IS EMPTY' kunnen voorkomen (als Q een wachtrij is). De selectie van een bepaalde component uit een queue of een verzameling is een krachtig hulpmiddel. Bijvoorbeeld:

```
FIRST PERSON IN ROOM THAT BELONGS TO CLUB.
```

Een continu attribuut van een component wordt gedeclareerd bijvoorbeeld als CONTINUOUS (2): DIST. Dit leidt tot drie attributen van het type REAL: DIST, DIST' en DIST"; deze kunnen in ieder statement worden gebruikt en van waarde veranderen door een assignment. De integratie van alle continue variabelen vindt plaats volgens een voorschrift als:

```
SPECIFY DIST PRECEPT(DIST" < 1) INIT (0,0)
ABSERROR (0.01) RELError (0.01)
```

De versnelling, DIST" wordt constant op 1 gehouden; de initiële waarden van DIST en DIST' zijn 0 en 0 en de fouten beide 0.01. Zowel discontinuïteiten als state-events (gebeurtenissen die optreden als een bepaalde toestand optreedt; zie WAIT WHILE conditie) worden ontdekt.

Een macro/procedure wordt in de component genoemd of aangeroepen met CALL SELECT en bevat een serie opdrachten die eindigen met RETURN.

Input geschiedt met READ voor integers en reals en met CHREAD voor character strings. Beide kunnen in expressies voorkomen. Beide kunnen ook worden gevolgd door een verwijzing naar een file: READ FROM FL. Het handige van zo'n verwijzing FL naar een file is dat PROSIM het mogelijk maakt om buiten het model om FL te verbinden met een concrete datafile of voor verscheidene runs met een reeks van datafiles (zie Maintenance system).

Een middel om de structurering te bevorderen is om bijvoorbeeld de initiële input in een aparte macro te regelen. Voor de output kan een aparte component worden gemaakt die de nodige waarnemingen verricht. Zo wordt een scheiding tussen model en experiment nagestreefd. Een dergelijke scheiding is overigens niet altijd mogelijk; als er niet regelmatig in de tijd wordt waargenomen maar zo nu en dan bijvoorbeeld juist voor en na een verandering in een component dan is die opdracht in de component zelf nodig.

Maintenance System.

Het maintenance system regelt:

- catalogus van models en datafiles;
- creëren/verwijderen en veranderen van models (modules/macro's) en datafiles en het importeren van modules/macro's uit andere modellen;
- printen van modules/macro's en datafiles;
- backup en restore complete models en datafiles (incl. environments);
- vertalen van modules/macro's;
- organiseren van simulatie-environments.

Er wordt een onderscheid gemaakt tussen modellen en datafiles. Dit heeft twee gevolgen. Iedere datafile kan worden gebruikt door ieder model. Bovendien kan de output file van een model input file zijn voor een ander model. Om model en datafiles aan elkaar te verbinden wordt een simulatie environment gehanteerd. Verscheidene environments bij een model stellen dan afzonderlijke experimenten voor. Een environment bestaat uit een lijst van inputstream reference attributes met daarachter de datafile naam of screen en een lijst van outputstream references attributes van het model met daarbij de naam van een datafile of screen of printer.

Naast de user datafiles (voor input of output) bestaan er store files en figure files (beide voor output: grafieken resp. animatie). Deze files kunnen aan het einde van de simulatie al dan niet worden bewaard. De store file is ook beschikbaar om tijdens de simulatie al grafieken van een variabele tegen de tijd te maken.

Met prosim/b = filename kan een batch file met een reeks van modellen en environments worden afgewerkt. Ook is het mogelijk om een model in een andere shell op te nemen en op een andere PC te draaien.

Run Control System.

In het Run Control System kan de gebruiker een simulatie run onderbreken door Esc in te drukken, door een geprogrammeerde INTERRUPT of door een breakpoint te zetten. Aan het begin van een run is er automatisch al zo'n interrupt, waardoor de volgende acties mogelijk zijn:

- gebruik een TRACE: acties van componenten op de tijdas en in wachtrijen verschijnen op het scherm;
- een breakpoint in de simulatietijd aanbrengen;
- de methode van integratie van de continue variabelen kan worden ingesteld;
- data handling: grafieken van variabelen in de store file; gegevens van wachtrijen;
- analyse van de toestand van een model: de waarde van variabelen is op te vragen en te veranderen (numerieke var.), het behoren van componenten tot een wachtrij is te achterhalen, ook die van de tijdas (EVENT CHAIN);
- voortzetten of stoppen van de simulatie run.

Supplements.

Bij PROSIM behoren drie supplements die van nut zijn na de simulatie: de animatie, de grafieken en de statistiek.

- **Animation**

Het dynamisch gedrag van een systeem visueel presenteren is nuttig om uit te leggen hoe een model werkt of om de kracht van een model in kritische situaties te laten zien en indien cijfers over een gehele run te weinig zeggen. Daarnaast kan het ook nog worden benut om te verifiëren of het model doet wat wordt beoogd.

- **Graphics**

Een tijdreeks van een variabele weergegeven in een plaatje of een histogram kan reeds toegepast worden tijdens een interrupt van een simulatie. Na de simulatie-run worden er nog enkele mogelijkheden toegevoegd:

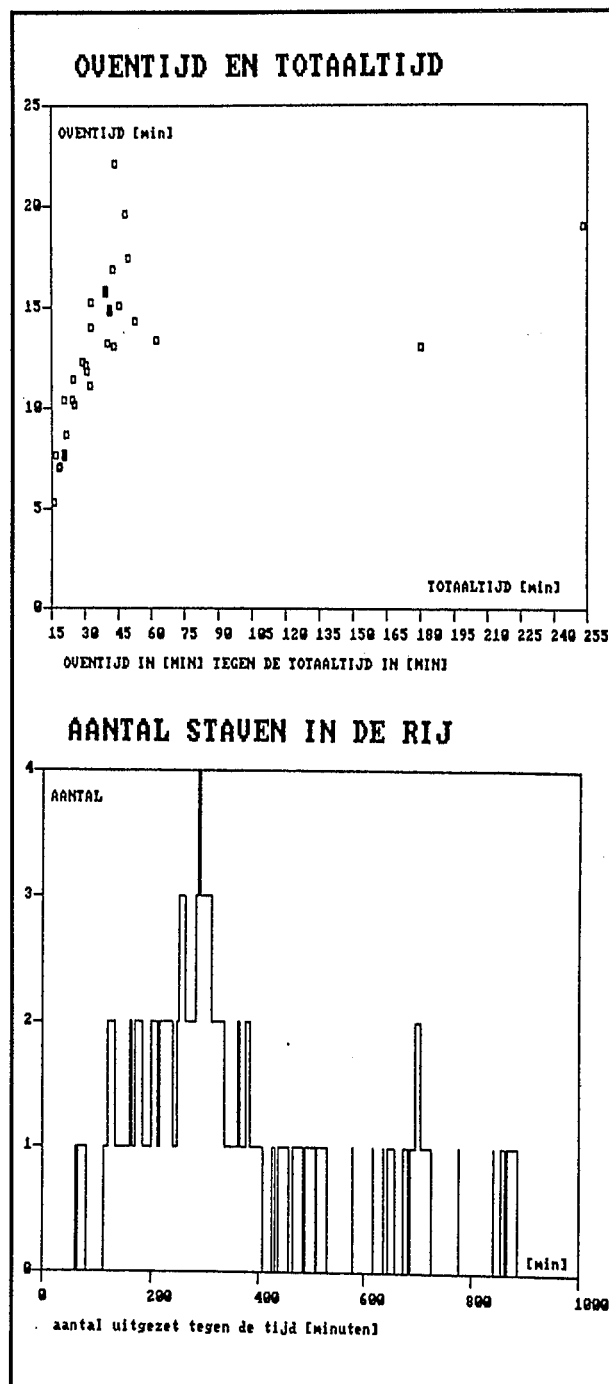
- het opnemen van tekst en figuurtjes in het plaatje;
- het samenvoegen van verscheidene plaatjes (max. 5) tot een grotere figuur;
- het samenvoegen van eenzelfde variabele uit verscheidene simulatie runs (max. 4) in één grafiek; ook waarnemingen van een tijdreeks in het reële systeem kunnen worden opgenomen;
- het tegen elkaar uitzetten van twee variabelen.

- **Statistics**

De tijdreeksen zijn hier eveneens opgeslagen in store files (simulatie output) of user data files (gedrag van reële systeem). Een tijdreeks kan worden opgedeeld en de delen kunnen afzonderlijk worden bestudeerd of met elkaar worden vergeleken. Ook een voorwaardelijke selectie uit een tijdreeks is mogelijk. De drie terreinen waarop dit supplement zich beweegt zijn de bewerking van één variabele, de vergelijking van twee variabelen en regressie.

- De bewerking van één variabele omvat o.a.: elementaire statistische gegevens, transformatie van de variabele, auto correlatie (om periodiciteit te ontdekken), percentielen, passen van de waarnemingen bij een bepaalde verdeling, histogram.
- Twee tijdreeksen met elkaar vergelijken: histogrammen, twee cumulatieve verdelingen, plot van de een tegen de andere variabele, test op gelijkheid van verdelingen, nieuwe reeks uit twee oorspronkelijke reeksen.
- Regressie.

Een lineaire benadering van de afhankelijke variabele en de parameters van zo'n meta-model worden berekend. Er wordt gewaarschuwd als de coëfficiënten lineair afhankelijk zijn van elkaar; zo niet dan wordt een variantie analyse of een chi-kwadraat



Figuur 1: Resultaat STORE opdracht

PERSONAL PROSIM RUNTIME SYSTEM - MODEL IS VIASVRBL

Gemiddelde tussen-aankomsttijd van de staven? 30

QUEUE NAME	NUMBER OF				INFORMATION ABOUT WAITINGTIMES		
	ENTRIES	ZERO ENTRIES	CURRENT LENGTH	MAX. LENGTH	MEAN	DEVIATION	MAXIMUM
OVEN	32	0	0	1	12.766	3.998	22.106
RIJ	32	5	0	4	22.029	46.055	225.250

Time = 960.00000

Figuur 2: Voorbeeld schermuitvoer

```

1 @ DEFINITIE GEDEELTE
2 @ DECLARATIES van MAIN
3 QUEUE          : RIJ  OVEN
4 RANDOMSTREAM   : INTERVAL BEWERK NORM
5 TIMEUNIT       : MINUTE
6
7 @ CLASS STAAF en ATTRIBUTES
8 CLASS          : STAAF
9 ATTRIBUTES OF STAAF:
10 CONTINUOUS(1) : TEMP
11 REAL          : OMGEVINGS_TEMP FACTOR
BEWERK_DUUR OVEN_TIJD
12
13 @ COMPONENT GENERATOR en ATTRIBUTES
14 COMPONENT     : GENERATOR
15 ATTRIBUTES OF GENERATOR:
16 REAL         : GEMIDD
17
18 @ COMPONENT MACHINE en ATTRIBUTES
19 COMPONENT    : MACHINE
20 ATTRIBUTES OF MACHINE:
21 REFERENCE TO STAAF : STAAF_IN_OVEN
22 REAL         : OVENTEMP

1 @ PROCESS van MAIN Initialisatie van MAIN en
CLASSES
2
3 SPECIFY TEMP PRECEPT(TEMP' FACTOR x
(OMGEVINGS_TEMP - TEMP)) INIT(300)
ABSERROR(0.1)
RELEERROR(0.1)
4
5 RESHAPE INTERVAL AS SAMPLED FROM
DISTRIBUTION EXPONENTIAL
6 RESHAPE BEWERK AS SAMPLED FROM DISTRIBUTION
NORMAL WITH PARAMETERS
MEAN(10) DEVIATION(2)
7 RESHAPE NORM AS SAMPLED FROM DISTRIBUTION
NORMAL WITH PARAMETERS
MEAN(0.1) DEVIATION(0.02)
8
9 @ PROCESS van MAIN Dynamic
10 ACTIVATE GENERATOR FROM MAAK
@ start van generator
11 ACTIVATE MACHINE FROM START IN MCHMOD
@ start van macine
12
13 WAIT 16 HOURS
@ lengte van de simulatieduur in uren!
14 CANCEL ALL
@ Alle wachtende processen worden verwijderd
15 PRINT STATISTICS
@ statistische gegevens van de queues
16 TERMINATE
@ beeindig de eigen activiteit; omdat er nu geen
processen meer aanwezig zijn stopt de simulatie!
17
18 @ PROCESS DESCRIPTION van de GENERATOR Initial
19 MAAK:
20 WRITE "Gemiddelde tussen-aankomsttijd van de
staven" WITH IMAGE

21 GEMIDD <- READ
22 @ Dynamic
23 VOLGENDE:
24 THIS STAAF<-NEW STAAF
25 FACTOR <- NORM
26 BEWERK_DUUR <- BEWERK
27 ACTIVATE THIS STAAF FROM BEGIN_STAAF
28 WAIT (INTERVAL x GEMIDD) MINUTES
@ tijd tussen aankomsten van staven
29 REPEAT FROM VOLGENDE
30
31 @ PROCESS DESCRIPTION van een STAAF Initial &
Dynamic
32 BEGIN_STAAF:
33 OMGEVINGS_TEMP <- 20
34 STORE LENGTH OF RIJ AS "R"
@ lengte voor
35 ENTER RIJ
36 STORE LENGTH OF RIJ AS "R"
@ lengte na opnemen van staaf in rij
37
38 INTEGRATE WHILE TEMP > 30
@ integratie daarna niet interessant
39 WAIT WHILE OMGEVINGS_TEMP < 100
@ zolang nog niet in de oven opgenomen
40 TEMP <- 25 IF TEMP < 31
@ keuze van temp. tussen 20 en 30
41 INTEGRATE
@ einde wordt in de machine geregeld

1 @ PROCESS DESCRIPTION van de MACHINE Dynamic
2 START:
3 OVENTEMP <- 500
4 WAIT WHILE RIJ IS EMPTY
5 STAAF_IN_OVEN <- FIRST STAAF IN RIJ WITH
GREATEST TEMP @ <- efficient!
6 STORE LENGTH OF RIJ AS "R" @ lengte voor
7 REMOVE STAAF_IN_OVEN FROM RIJ
8 STORE LENGTH OF RIJ AS "R"
@ lengte na het verwijderen van bar
9 JOIN STAAF_IN_OVEN TO OVEN
10 OVEN_TIJD OF STAAF_IN_OVEN <- NOW
@ legt start van verblijf in oven vast
11 OMGEVINGS_TEMP OF STAAF_IN_OVEN <- OVENTEMP
12 WAIT WHILE TEMP OF STAAF_IN_OVEN < 400
@ tot staaf is opgewarmd
13 REMOVE STAAF_IN_OVEN FROM OVEN
14 OMGEVINGS_TEMP OF STAAF_IN_OVEN <- 20
@ nu bar weer in de buiten temp.
@ verblijfsduur in de oven
15 OVEN_TIJD OF STAAF_IN_OVEN <- NOW - OVEN_TIJD
OF STAAF_IN_OVEN
16 WORK BEWERK_DUUR OF STAAF_IN_OVEN
@ staaf kan worden bewerkt
17 STORE OVEN_TIJD OF STAAF_IN_OVEN AS "O"
@ totale tijd staaf wachten + verwarmen + bewerken
18 STORE NOW - ARRIVALTIME OF STAAF_IN_OVEN AS "P"
@ job stopt met integratie
19 CANCEL STAAF_IN_OVEN
20 REPEAT FROM START

```

Figuur 1: Programmacode oven-voorbeeld

waarde gepresenteerd. Vervolgens worden de schattingen van de parameters, de standaard afwijking en het 95% betrouwbaarheids interval van deze parameters gegeven. Tenslotte wordt een diagram vertoond van de oorspronkelijke waarde van de afhankelijke variabele uitgezet tegen de berekende waarde.

Voorbeeld

Het model 'VIASVRBL' laat met 'GENERATOR' hete staven 'STAAF' van 300 °C in een wachtrij 'RIJ' komen waar ze door de omgevingstemperatuur van 20 °C afkoelen tot beneden een temperatuur van 30 °C, tenzij ze voordien worden uitgekozen om in de 'OVEN' van 500 °C te worden verwarmd tot 400 °C om vervolgens door een 'MACHINE' te worden bewerkt. Om niet alle staven op te nemen in de integratie wordt een oninteressant deel (20 - 30 °C) niet uitgevoerd.

De resultaten van de PRINT STATISTICS hebben betrekking op de twee queues 'OVEN' en 'RIJ' en zijn een

gevolg van de gemiddelde tussen-aankomst tijd van staven in de 'GENERATOR' van 'GEMIDD' 30 minuten (opgegeven op het scherm na de betreffende vraag).

De figuren zijn een resultaat van de STORE opdrachten en via het Supplement GRAPHICS omgevormd tot een plaatje met bijbehorende tekst.

Literatuur

N.N., 1988, *Personal Prosim, Textbook, Reference Manual, Supplements Sierenberg & de Gans*, Wadinxveen, 1988, 162 p.+ 129p.+(26+16+33p.)

Wit, C.T. de, 1978, *Simulatie van levende systemen*. Landbouwkundig Tijdschrift, p. 237-240. □

Dr. ir. E. van Elderen is werkzaam bij het IMAG, Postbus 43, 6700 AA Wageningen, Mansholtlaan 10-12, 6708 AA Wageningen, tel. 08370-94324, telefax 08370-94666.