

DISCRETE SIMULATIE  
=====

Ir. W.C. Boot  
Prof. Dr. Ir. J.E. Rooda

## 1.0 SAMENVATTING

Om een systeem te kunnen simuleren is het nodig om over een model van dat systeem te beschikken. Aan de hand van een eenvoudig voorbeeld wordt de bouw van dergelijke modellen volgens de Proces Interactie benadering toegelicht.

## 2.0 INLEIDING

Discrete simulatie is een techniek die vooral toepassing vindt bij onderzoek naar het gedrag van bestaande of nog te bouwen discrete systemen. Karakteristiek voor dit soort systemen is dat het gedrag in principe kan worden beschreven als een reeks gebeurtenissen. Bij elke gebeurtenis verandert de toestand van het systeem. Tussen twee gebeurtenissen blijft de toestand gelijk. Discrete simulatie is vooral een interessante techniek bij onderzoek aan grote systemen, waarbij de gebeurtenissen een stochastisch karakter hebben.

Om een simulatie te kunnen uitvoeren is het noodzakelijk dat eerst een model van het systeem wordt gebouwd. Dit dient een zodanige vorm te hebben dat het zich leent voor omzetting in een op een computer uitvoerbaar programma.

Een zeer bruikbare methode hiervoor is de Proces Interactie benadering gebleken. Deze maakt op relatief eenvoudige en inzichtelijke wijze de afbeelding van een systeem op een model mogelijk. Voor andere benaderingen wordt verwezen naar Nance (1981).

Deze tekst beoogt enig inzicht te verschaffen in de wijze waarop deze afbeelding tot stand komt. Daartoe worden de basisbegrippen van de Proces Interactie benadering gepresenteerd aan de hand van een eenvoudig voorbeeld. Hierbij zal gebruik worden gemaakt van het in Modular Pascal [Bron 1982] geschreven simulatiepakket S84 [Rooda, Joosten, Rossingh, Smedinga 1984].

-----

De auteurs zijn werkzaam aan de afdeling Werktuigbouwkunde van de T.H. Eindhoven

### 3.0 BESCHRIJVING VAN HET SYSTEEM

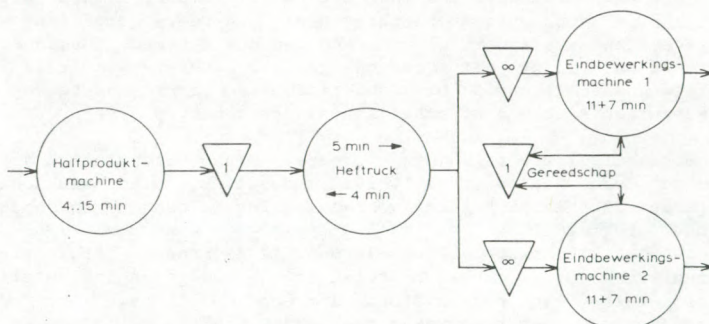
In het beschouwde systeem wordt een aantal verschillende halfprodukten vervaardigd door een volautomatische machine. Deze machine is zeer flexibel en kan alle types zonder omstellen direkt na elkaar vervaardigen. Dit duurt, afhankelijk van het type, 4 tot 15 minuten. Aangenomen wordt dat de aanvoer van grondstoffen geen problemen geeft.

Zodra deze machine klaar is plaatst hij het halfprodukt in een kleine buffer. Hierin is slechts plaats voor 1 exemplaar van het halfprodukt. Bij plaatsgebrek in de buffer zal de machine het zojuist vervaardigde halfprodukt niet kunnen afleveren. Daardoor kan ook niet worden begonnen met de vervaardiging van het volgende halfprodukt.

Vanuit de buffer wordt het halfprodukt door een heftruck getransporteerd naar een machine die de eindbewerking uitvoert. Er zijn twee van deze machines. De heftruck brengt de halfprodukten afwisselend naar de ene of de andere machine. Voor een rit met een halfprodukt heeft de heftruck 5 minuten nodig. De duur van de terugrit is een minuut korter.

De twee machines voor de eindbewerking beschikken beide over een buffer voor te bewerken halfprodukten. Voor praktische gevallen kan deze als onbegrensd worden beschouwd. De duur van de bewerking is 18 minuten, ongeacht het type produkt. Gedurende de laatste 7 minuten van de bewerking is een gereedschap nodig waarvan slechts 1 exemplaar aanwezig is. Het is dus onmogelijk dat beide machines tegelijk aan het laatste deel van de bewerking bezig zijn. De afvoer van gereede produkten geschiedt probleemloos.

De volgende figuur geeft de situatie schematisch weer:



### 4.0 HET MODEL.

In het systeem is een viertal elementen te onderscheiden dat acties uitvoert. Dit zijn respectievelijk de halfprodukt machine, de heftruck en de twee eindbewerkingsmachines. Voor het beschrijven van het gedrag van het model is het grootste probleem dat de acties van de verschillende elementen onafhankelijk van elkaar in de tijd plaatsvinden.



De voortgang van de verschillende acties wordt slechts beïnvloed door de toestand van een viertal andere elementen. Dit zijn de buffer na de halfprodukt-machine, de buffers voor de eindbewerkingsmachines en het gereedschap dat door deze machines samen wordt gebruikt.

De eerst genoemde buffer kan zowel de voortgang van de halfprodukt-machine beïnvloeden als die van de heftruck. De machine kan geen produkt kwijt wanneer de buffer nog vol is, terwijl de heftruck moet wachten wanneer de buffer leeg is.

De twee eindbewerkingsmachines kunnen niet verder wanneer hun respectievelijke buffers leeg zijn. Ook kan de voortgang van de ene machine worden belemmerd wanneer het gemeenschappelijke gereedschap in gebruik is door de andere machine.

De Proces Interactie benadering biedt middelen om de onafhankelijk in de tijd functionerende elementen te beschrijven. Deze beschrijvingen worden aangeduid met de naam PROCESS. Tevens kunnen elementen worden beschreven (de buffers) waarvan de toestand de voortgang in het systeem bepaalt. Zij bepalen (via een omweg) de interactie tussen de diverse processen.

#### 4.1 De beschrijving van een Proces.

In S84 wordt een proces als volgt beschreven:

```
PROCESS Procesnaam (procesparameters);
(* locale declaraties *)
BEGINPROC
  (* acties van het proces *)
ENDPROC;
```

In de beschrijving van de acties van een proces zijn (mogelijk) drie verschillende operaties aan te treffen die karakteristiek zijn voor processen. Dit zijn:

- Het gedurende een bepaalde tijd bezig zijn met een activiteit. Dit wordt aangegeven met "work (tijdsduur)".
- Het zetten van een produkt in een buffer. Dit wordt aangegeven met "v (buffer)".
- Het nemen van een produkt uit een buffer. Hierbij kan een complicatie ontstaan, namelijk dat de buffer leeg is. In dat geval zal het proces dienen te wachten tot er weer wat in de buffer wordt gezet. Deze combinatie van inspecteren van de buffer, er zo mogelijk iets uitnemen en anders wachten wordt aangegeven door "p (buffer)".

De genoemde v- en p-operaties ontleen hun naam aan de informatica. Het zijn daar de elementaire operaties op zogenaamde semaphoren [Dijkstra 1968].

Een element dat de acties uit een bepaalde procesbeschrijving kan uitvoeren wordt gecreeerd door aanroep van de procedure activate. Op deze wijze is het eenvoudig mogelijk meerdere elementen met dezelfde beschrijving te maken. Ze verschillen slechts in de eventuele procesparameters. Dit ziet er als volgt uit:

```
activate (procesnaam (^elementnaam^, procesparameters), delta);
```

Hierbij geeft delta aan na hoeveel tijd het proces wordt geactiveerd, dat wil zeggen dat op dat moment de uitvoering van de procesbeschrijving begint.

#### 4.2 De beschrijving van een buffer

Buffers met onbegrensde capaciteit zijn zonder meer beschikbaar. Deze worden aangegeven als variabelen van het type "sema":

```
VAR buffer: sema
```

Buffers met een bovengrens kunnen worden opgebouwd uit twee buffers, beide met onbegrensde capaciteit. De eerste geeft de feitelijke inhoud van de buffer weer, de andere de hoeveelheid beschikbare ruimte. Iets in de buffer zetten ziet er dan uit als p(ruimte), gevolgd door v(buffer). Analooog wordt iets uit de buffer gehaald door achtereenvolgens p(buffer) en v(ruimte).

Een buffer wordt geïnitieerd met behulp van de procedure iniSema. Hierbij krijgt de buffer een naam en een begin-inhoud. Dit gebeurt als volgt:

```
iniSema (buffer, 'buffernaam', beginInhoud)
```

#### 5.0 HET MODEL GEPROGRAMMEERD

Met bovengenoemde middelen kunnen de verschillende elementen van het model in beginsel worden geprogrammeerd. De halfprodukt-machine ziet er dan bijvoorbeeld als volgt uit:

```
PROCESS HalfproduktMachine;
BEGINPROC
  LOOP work( (* 4 tot 15 minuten *) );
    p(halfproduktRuimte);
    v(halfproduktBuffer)
  END
ENDPROC;
```

In deze declaratie ontbreekt nog het stochastische karakter van de cyclusduur van de halfprodukt-machine. Ook hiervoor stelt S84 middelen ter beschikking. De functie realSample(distr) levert een trekking uit de door de variabele distr gespecificeerde distributie. De variabele distr is van het TYPE dist. Bij initialisatie van een variabele van dit type wordt de aard van de distributie opgegeven, alsmede de karakteristieke parameters. Een uniforme verdeling tussen 4 en 15 wordt als volgt gemaakt:



```

(* program heading *)
VAR distr: dist;
(* andere declaraties *)
BEGIN
  iniDist(distr, uniform, 4, 15);
  (* andere statements *)
END.

```

Nu kan het volledige model worden opgesteld, inclusief alle initialisaties en activeringen:

```

PROGRAM fabriek (s84def, s84);

VAR halfproduktBuffer : sema;
    halfproduktRuimte : sema;
    eindbeworkBuffer : ARRAY [0..1] OF sema;
    gereedschap       : sema;

PROCESS HalfproduktMach;
VAR cyclus: dist;
BEGINPROC
  iniDist(cyclus, uniform, 4, 15);
  LOOP
    work(realSample(cyclus));
    p(halfproduktRuimte); v(halfproduktBuffer)
  END
ENDPROC;

PROCESS Heftruck;
VAR bestemming: integer;
BEGINPROC
  bestemming:= 0;
  LOOP
    p(halfproduktBuffer); v(halfproduktRuimte);
    work(5);
    v(eindbeworkBuffer[bestemming]);
    bestemming:= 1 - bestemming;
    work(4);
  END
ENDPROC;

PROCESS eindbeworking(nr: integer);
BEGINPROC
  LOOP
    p(eindbeworkBuffer[nr]);
    work(11);
    p(gereedschap);
    work(7);
    v(gereedschap)
  END
ENDPROC;

```

```

BEGIN
  iniS84;
  iniSema(halfproduktBuffer , 'HalfprBuf' , 0);
  iniSema(halfproduktRuimte , 'HalfprRmt' , 1);
  iniSema(gereedschap , 'Gereedschap' , 1);
  iniSema(eindbewerkBuffer[0] , 'EindBuf 1' , 0);
  iniSema(eindbewerkBuffer[1] , 'EindBuf 2' , 0);
  activate(HalfproduktMach('H.ProdMach'), 0);
  activate(Heftruck('Heftruck') , 0);
  activate(eindbewerking('EindBew 1', 0), 0);
  activate(eindbewerking('EindBew 2', 1), 9);
  hold(100) (* simulatieduur *)
END.

```

## 6.0 GEGEVENS OVER HET SYSTEEMGEDRAG

In het beschreven programma ontbreekt nog iedere uitvoer. Executie van het programma zal dan ook weinig inzicht geven in het gedrag van het model, want er is niets aan te zien. Hierin kan verandering worden gebracht door bijvoorbeeld het simulatie programma een zogenaamde trace te laten uitvoeren. De trace bevat voor elke gebeurtenis in het model een regel tekst. Deze bestaat uit een vermelding van de tijd en een beschrijving van de betreffende gebeurtenis.

Voor een bepaalde run van het programma volgt nu de trace. Omdat deze nogal veel ruimte in beslag neemt is tussen tijdstip 51.0 en 98.0 een stuk weggelaten.

```

*****
*                               Clock time =      +0.000                               *
*                               Tracing Starts                                         *
*****

```

| Time    | Current    | Action                                                                                                                                                          |
|---------|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| +0.000  | S84        | activates H.ProdMach at once<br>activates Heftruck at once<br>activates EindBew 1 at +22.000<br>activates EindBew 2 at +22.000<br>holds +100.000 until +100.000 |
|         | H.ProdMach | holds +6.972 until +6.972                                                                                                                                       |
| +6.972  | Heftruck   | waits in a P operation on HalfprBuf                                                                                                                             |
|         | H.ProdMach | does a P operation on HalfprRmt<br>does a V operation on HalfprBuf<br>holds +11.502 until +18.474                                                               |
|         | Heftruck   | leaves the P operation on HalfprBuf<br>does a V operation on HalfprRmt<br>holds +5.000 until +11.972                                                            |
| +11.972 |            | does a V operation on EindBuf 1<br>holds +4.000 until +15.972                                                                                                   |
| +15.972 |            | waits in a P operation on HalfprBuf                                                                                                                             |
| +18.474 | H.ProdMach | does a P operation on HalfprRmt<br>does a V operation on HalfprBuf<br>holds +11.534 until +30.008                                                               |
|         | Heftruck   | leaves the P operation on HalfprBuf<br>does a V operation on HalfprRmt<br>holds +5.000 until +23.474                                                            |



```

+22.000 EindBew 1    does a P operation on EindBuf 1
                    holds +11.000 until +33.000
                    EindBew 2    waits in a P operation on EindBuf 2
+23.474 Heftruck     does a V operation on EindBuf 2
                    holds +4.000 until +27.474
                    EindBew 2    leaves the P operation on EindBuf 2
                    holds +11.000 until +34.474
+27.474 Heftruck     waits in a P operation on HalfprBuf
+30.008 H.ProdMach   does a P operation on HalfprRmt
                    does a V operation on HalfprBuf
                    holds +13.893 until +43.901
                    Heftruck     leaves the P operation on HalfprBuf
                    does a V operation on HalfprRmt
                    holds +5.000 until +35.008
+33.000 EindBew 1    does a P operation on Gereedschap
                    holds +7.000 until +40.000
+34.474 EindBew 2    waits in a P operation on Gereedschap
+35.008 Heftruck     does a V operation on EindBuf 1
                    holds +4.000 until +39.008
+39.008              waits in a P operation on HalfprBuf
+40.000 EindBew 1    does a V operation on Gereedschap
                    does a P operation on EindBuf 1
                    holds +11.000 until +51.000
                    EindBew 2    leaves the P operation on Gereedschap
                    holds +7.000 until +47.000
+43.901 H.ProdMach   does a P operation on HalfprRmt
                    does a V operation on HalfprBuf
                    holds +8.830 until +52.731
                    Heftruck     leaves the P operation on HalfprBuf
                    does a V operation on HalfprRmt
                    holds +5.000 until +48.901
+47.000 EindBew 2    does a V operation on Gereedschap
                    waits in a P operation on EindBuf 2
+48.901 Heftruck     does a V operation on EindBuf 2
                    holds +4.000 until +52.901
                    EindBew 2    leaves the P operation on EindBuf 2
                    holds +11.000 until +59.901
+51.000 EindBew 1    does a P operation on Gereedschap
                    holds +7.000 until +58.000
                    ...          -----/-----
+98.431 H.ProdMach   does a P operation on HalfprRmt
                    does a V operation on HalfprBuf
                    holds +4.857 until +103.288
                    Heftruck     leaves the P operation on HalfprBuf
                    does a V operation on HalfprRmt
                    holds +5.000 until +103.431
*****
*                      Clock time =  +100.000                      *
*                      Tracing Stops                               *
*****

```

Zoals blijkt beslaat een trace veel tekst, terwijl het nog steeds moeilijk is om de zinvolle informatie er uit te halen. Daarom bestaat er behoefte aan meer doelgerichte methoden voor het verzamelen van gegevens. Van de verschillende mogelijkheden die S84 biedt zal hier het histogram worden gebruikt.

Evenals de andere voorzieningen voor het verzamelen van gegevens is een histogram een variabele van het TYPE dcd (Data Collection Device). Deze wordt geïnitialiseerd met behulp van de procedure

```
iniHisto(histogram, 'naam', ondergrens, bovengrens, aantalKlassen)
```

Een waarneming wordt aan een histogram toegevoegd met de procedure

```
update(histogram, waarneming)
```

Het histogram zal worden gebruikt om bij te houden hoe vaak en hoe lang een eindbewerkingsmachine moet wachten bij gebrek aan halfprodukten. Dit vereist de volgende aanpassingen in het programma:

- De declaratie van de VAR geenHalfPr: dcd.

- De aanpassing van PROCESS eindbewerking:

```
PROCESS eindbewerking(nr: integer);
VAR t: real;
BEGINPROC
  LOOP t:= time; p(eindbewerkBuffer[nr]); t:= time - t;
    IF t > 0.01 THEN update(geenHalfPr, t);
      work(11);
      p(gereedschap); work(5); v(gereedschap)
    END
  ENDPROC;
```

Het IF statement voorkomt het steeds registreren van het geval dat er geen noemenswaardige wachttijd optreedt.

- Het initialiseren van het histogram, bijvoorbeeld:

```
iniHisto(geenHalfPr, 'geen HalfPr', 0, 7, 7).
```

Bij een simulatieduur van 400 minuten werd het volgende histogram geregistreerd:

| Title          | Obs | Reset at | Average | Est.st.dv | Minimum | Maximum |
|----------------|-----|----------|---------|-----------|---------|---------|
| geen Halfpr    | 19  | +0.000   | +2.393  | +1.695    | +0.431  | +5.849  |
| Cell Lower lim | N   | Freq     | Cum %   | I         | -----   |         |
| 0 -Infinity    | 0   | +0.00    | +0.00   | I         |         |         |
| 1 +0.000       | 5   | +0.26    | +26.32  | I*****    |         |         |
| 2 +1.000       | 7   | +0.37    | +63.16  | I*****    |         |         |
| 3 +2.000       | 1   | +0.05    | +68.42  | I****     |         |         |
| 4 +3.000       | 2   | +0.11    | +78.95  | I*****    |         |         |
| 5 +4.000       | 2   | +0.11    | +89.47  | I*****    |         |         |
| 6 +5.000       | 2   | +0.11    | +100.00 | I*****    |         |         |
| 7 +6.000       | 0   | +0.00    | +100.00 | I         |         |         |
| 8 +7.000       | 0   | +0.00    | +100.00 | I         |         |         |
|                |     |          |         | I-----    |         |         |

Het blijkt dat er 19 keer een wachttijd van meer dan 0.01 minuut is opgetreden. Deze duurde dan gemiddeld 2.393 minuten met een minimum van 0.431 minuut en een maximum van 5.849 minuut.

Uit deze gegevens is de gemiddelde wachttijd per cyclus niet op eenvoudige wijze af te leiden, daar het aantal cycli onbekend is. Door bijvoorbeeld het bijwerken van het histogram onvoorwaardelijk te doen kan dit gegeven wel worden bepaald.



## 7.0 SLOTOPMERKING

Bij de uitwerking van het voorbeeld is nauwelijks aandacht besteed aan allerlei problemen die de kop op steken bij het modelleren van praktische systemen. Ook zijn van de vele mogelijkheden die een pakket als S84 biedt slechts de meest elementaire aan de orde gekomen. Deze onderwerpen zijn zeker niet onbelangrijk. Aandacht besteden hieraan zou echter gemakkelijk de aandacht te veel hebben afgeleid van de kern van de problematiek.

## 8.0 LITERATUUR

Bron C.,  
Modular Pascal Language Definition,  
Memorandum nr. INF-82-10,  
TH Twente, afd. Informatica, Enschede (1982).

Dijkstra E.W.,  
Cooperating Sequential processes (1965),  
reprinted in Programming Languages (ed. F. Genuys),  
Academic Press, New York NY (1968).

Nance R.E.,  
The time and state relationships in simulation modelling,  
Communications of the ACM 24 (4), 173-179 (1981).

Rooda J.E., Joosten S.M.M., Rossingh T.J., Smedinga R.,  
Discrete simulation in S84,  
Memorandum nr. 40  
TH Twente, afd. werktuigbouwkunde, Enschede (1984).