

Instituut voor Cultuurtechniek en Waterhuishouding  
Wageningen

## ASPECTEN van INFORMATIEVERWERKING

38

Het omzetten van Cyber-Fortranprogramma's naar  
Vax-Fortranprogramma's

Een snelle en eenvoudige methode

K. Oostindie

Nota's van het Instituut zijn in principe interne communicatiemiddelen, dus geen officiële publikaties.

Hun inhoud varieert sterk en kan zowel betrekking hebben op een eenvoudige weersave van cijferreeksen, als op een concluderende discussie van onderzoeksresultaten. In de meeste gevallen zullen de conclusies echter van voorlopige aard zijn omdat het onderzoek nog niet is afgesloten.

Bepaalde nota's komen niet voor verspreiding buiten het Instituut in aanmerking.

16 JUNI 1983

JSN107020-02

## ASPECTEN VAN INFORMATIEVERWERKING

38

De nota's handelend over Aspecten van Informatieverwerking bevatten inlichtingen over de ontwikkelings van de informatieverwerking binnen het Instituut. Naast meer concluderende en toelichtende beschouwingen wordt aandacht besteed aan het gebruik van programma's, programmapakketten en apparatuur. Tevens worden inlichtingen gegeven over praktijkervaringen met en toepassingen van de informatieverwerking.

## INHOUD

|                                                               | Pag. |
|---------------------------------------------------------------|------|
| 1.0 INLEIDING . . . . .                                       | 1    |
| 2.0 HET MEDIUM . . . . .                                      | 1    |
| 3.0 HET AANBIEDEN VAN TAPES AAN DE VAX-11 COMPUTER . . . . .  | 2    |
| 4.0 HET OVERZETTEN VAN EEN ENKEL PROGRAMMA . . . . .          | 3    |
| 5.0 VERSCHILLEN TUSSEN CYBER-FORTRAN EN VAX-FORTRAN . . . . . | 4    |
| 5.1 De PROGRAM-opdracht . . . . .                             | 5    |
| 5.2 Het lezen van en schrijven naar een terminal . . . . .    | 7    |
| 5.3 Formaat specificaties . . . . .                           | 7    |
| 5.4 Het dollarteken . . . . .                                 | 8    |
| 5.5 Het gelijkteken . . . . .                                 | 8    |
| 5.6 Het CHARACTER datatype . . . . .                          | 9    |
| 5.7 Test op end-of-file . . . . .                             | 9    |
| 5.8 Hulafuncties . . . . .                                    | 10   |
| 5.9 Hulroutines . . . . .                                     | 10   |
| 6.0 SLOTOPMERKING . . . . .                                   | 13   |
| REFERENTIES . . . . .                                         | 13   |
| BIJLAGE 1.                                                    |      |
| BIJLAGE 2.                                                    |      |

## 1.0 INLEIDING

In deze nota zal enige aandacht worden besteed aan het converteren van Cyber programma's naar Vax-11 programma's. De problemen die hier naar voren komt zijn: op welk medium moeten de files worden aangeleverd, hoe moeten de files aan de Vax-11 computer worden aangehouden en tenslotte, wat zijn de verschillen tussen Cyberfortran en Vaxfortran. In het kort zal op elk van deze drie problemen worden ingegaan. Tevens zal aan de hand van enige voorbeelden de gang van zaken rond de conversie worden verduidelijkt. De voorbeelden zijn toegespitst op de werksituatie bij het ICW zodat nader zal worden hoe de conversie verloopt van de Cyber van IWIS-TNO, den Haag, naar de Vax van het Staringgebouw, Wageningen.

## 2.0 HET MEDIUM

Het gekozen medium voor het transport tussen twee computers van bestanden en programma's is nog altijd de magnetische tape. Van dit medium zal dan ook gebruik worden gemaakt om de conversie te verzorgen. Er kan gebruik worden gemaakt van zogenaamde huurtapes van het IWIS-TNO in Den Haag of van eigen tapes, ook wel "strandertapes" genaamd. Het is in beide gevallen noodzakelijk hierover vooraf contact op te nemen met het productie bureau van het IWIS in den Haag. Via een batchjob kunnen de gewenste files vanaf permanent file worden gelezen en naar tape worden geschreven. Voorbeeld 1 geeft een voorstelling van zo'n batchjob.

De parameters in voorbeeld 1 welke vrij te kiezen zijn, zijn in kleine letters geschreven. In de COPYMF opdracht worden eerst de initialen van de eisenaar van de permanent-file gegeven en daarna de permanent-filename. Verder wordt aan deze opdracht een Z-parameter toegevoegd, hetgeen betekent dat de betreffende file in display-code staat. In geval de file in ASCII-code staat moet er in plaats van een Z- een A-parameter worden toegevoegd. Het is dus alleen de COPYMF-opdracht welke moet worden aangepast. Alle permanent-files worden tot 1 file samengevoegd en als 1 file op tape gezet. Alle records in de file krijgen een vaste lengte van 80 tekens. Hetgeen betekent dat de records eventueel met blanks worden aangevuld tot de vereiste recordlengte wordt verkregen. Deze blanks noemt men ook wel "trailing blanks". Door aanpassing van de parameter A=10\*80 in de COPYA-opdracht kan de recordlengte worden aangepast. (bijv: A=10\*132 geeft een vaste recordlengte van 132).

**Voorbeeld 1. Het op tafe zetten van permanent files.**

```

BIit1,T0,CM60000,NT1.                                (evt. DYit1,NT1.)
ACCOUNT,43,xxxxuzzzz,**.                               (volgens de geldende regels in te vullen)
ATTACH,ACL,ACCULIB.
LIBRARY,ACL.
COPYBR,INPUT,COPYMF.
REQUEST,MFN,NT,PE,S,MF,U,US,RING,VSN=it1.    TAPE VAN DHR/MW ... OPHANGEN
COMMENT. HERE COPYMF CALLS
COPYMF,it1,programma1,Z.
COPYMF,it1,databestand,Z.
UNLOAD,MFN.
.PROC,COPYMF,OWNER,PFN,FI.                             ( standaard procedure voor het
ATTACH,LFN,PFN,ID=OWNER.                                ( op tape zetten van files.
LABEL,TFN,W,M=MFN,L=PFN.                                ( alle files worden als 1 file
COPYA,I=LFN/FI,O=TFN/A=10*80,NOERR.                    ( op tape gezet.
UNLOAD,LFN,TFN.

```

### 3.0 HET AANBIEDEN VAN TAPES AAN DE VAX-11 COMPUTER

De nu beschreven tape mag naar het ICW worden opgestuurd en kan dan op de VAX-computer worden uitgelezen. Hoe dit in z'n werk gaat staat in voorbeeld 2 beschreven.

**Voorbeeld 2. Het van tape halen van files met behulp van de Vax-computer.**

[illegible]

In de COPY-opdracht moet de eisen directory worden gegeven. Filenaam en filetype kunnen vrij worden gekozen. Men dient er wel acht op te geven dat vooraf eerst het randapparaat wordt gereserveerd (ALLOCATE) en men aan de operator te kennen geeft welke tape moet worden opgehangen (REQUEST) en dat het opslagsmedium beschikbaar wordt gemaakt als 'stranger-tape' (MOUNT/FOR/OVERRIDE=IDENTIFICATION). De optie '/OVERRIDE=IDENTIFICATION' betekent dat eventueel aanwezige labels op de tape worden overseslagen. Hierover staat meer in ICW-nota 1382 (Oostindie, 1982).

De 'trailing blanks' kunnen worden verwijderd met behulp van het programma TRAILING. Hoe dit programma werkt staat aangegeven in bijlage 1. Hierna kan de nieuwe file met behulp van een editor (EDT) worden uitgesplitst naar verschillende files.

#### 4.0 HET OVERZETTEN VAN EEN ENKEL PROGRAMMA

Wanneer men slechts een enkel programma wil overzetten vanaf de Cyber naar de Vax, dan staat er nog een andere mogelijkheid dan door middel van tapes de gebruiker ten dienste. Men moet de gewenste file dan in de output-wachtrij van het IWIS\_TNO plaatsen, doch vooraf moeten eerst enkele handelingen worden ondernomen om tot het gewenste resultaat te komen. Normaal wordt een file uit een output-wachtrij op een printer afgedrukt. Daar dit hier niet gewenst wordt moet dit aan de operator te kennen worden gegeven. Dit kan worden gedaan door aan het begin van de file een PM-opdracht (Print Message) te plaatsen. Ook aan het einde van de file moet dan een PM-opdracht worden geplaatst. In voorbeeld 3 wordt zo'n file in de output-wachtrij geplaatst. Zoals uit dit voorbeeld blijkt wordt na het attachen van de file, een COPYSBF opdracht gegeven naar een andere file. Daarna wordt de nieuwe file in SUEFI gezet, waarna de PM-opdrachten in de file worden gezet. Een PM opdracht moet in de eerste kolom beginnen en mag niet meer dan 30 tekens commentaar bevatten. Wanneer nu de file uit de output-wachtrij wordt gehaald, verschijnt op de console-terminal de PM-opdracht en stopt de file op de printer. De operator kan nu de nodige voorzieningen treffen om de file te transporteren naar de VAX en deze in de juiste directory te plaatsen en de juiste naam te geven (in het voorbeeld is dit de directory [COIN.29990630] en de filenaam ONE.FOR).

Voorbeeld 3. Het overzetten van een enkele file.

```
COMMAND- ATTACH,A,PROGRAMMA1,ID=OIN.  
COMMAND- COPYSBF,A,B  
COMMAND- SUEDI  
..E,B,S  
    10 LINES READ  
..1=PM VAX:[COIN.29990630]ONE.FOR  
..L,L  
    100=      END  
110=PM DIT WAS DE FILE  
..S,C,O,N  
    12 LINES SAVED  
..B  
COMMAND- REQUEST,C,*Q  
COMMAND- ROUTE,C,TID=VE,DC=PR,FID=BITAP  
COMMAND-
```

De parameters welke in het ROUTE commando worden gebruikt hebben de volgende betekenis:

C = outputfile uit SUEDI  
TID = terminal identification (VE)  
DC = Disposition Code is PRinter  
FID = File Identification (BITAP)

## 5.0 VERSCHILLEN TUSSEN CYBER-FORTRAN EN VAX-FORTRAN

De files welke vanaf tape of via de output-wachtrij (VE) op de gebruikers directory zijn gezet zijn nu vanaf de terminal direkt toesankelijk. Alvorens een poging te wagen de gewenste programma's te vertalen is het zinvol eerst enige correcties aan te brengen met behulp van de editor EDT. Dit, omdat het Cyber-fortran op een aantal punten verschilt van het Vax-fortran. Op de meest voorkomende verschillen zal nader worden ingesaaan.

### 5.1 De PROGRAM-opdracht

In Cyber-fortran worden met de PROGRAM opdracht de files gedefinieerd welke men tijdens het uitvoeren van een programma wenst te gebruiken. In Vax-fortran is het slechts toegestaan met de PROGRAM opdracht alleen de programmanaam te noemen. Voor het definiëren van files moeten de OPEN en CLOSE opdrachten worden gebruikt. In voorbeeld 4 wordt dit onderscheid weerspiegeld. De READ en WRITE opdrachten zonder unitnummer, zoals de Cyber deze kent voor het lezen en schrijven naar respectievelijk de files INPUT en OUTPUT, kent de Vax, gebruikt in deze betekenis, niet. Het is dus noodzakelijk altijd een unitnummer in deze opdrachten mee te geven.

Voorbeeld 4. Het verschil in definiëren van files op een Cyber- en op een Vax-computer

```
Cyber :  PROGRAM ONE(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT)
        .
        .
        .
        STOP
        END
```

---

```
Vax :    PROGRAM ONE
        .
        .
        .
        OPEN(UNIT=1,NAME='INPUT.DAT',TYPE='OLD') !OLD= een bestaande file
        OPEN(UNIT=2,NAME='OUTPUT.DAT',TYPE='NEW') !NEW= File moet worden
        .                                     ! gemaakt.
        .
        .                                     ! De files worden
        CLOSE(UNIT=1)                        ! afgesloten en
        CLOSE(UNIT=2)                        ! blijven bewaard.
        STOP
        END
```







Wanneer de tekst langer is dan 1 fortranregel en er derhalve gebruik wordt gemaakt van een vervolgsregel, dan moeten de eventuele spaties aan het einde van de eerste fortranregel opnieuw worden ingevoerd. Dit omdat de Cyber-computer de spaties aan het eind van een fortranregel afkapt, terwijl het fortranvertaalprogramma de spaties weer toevoegt. Door een Vax-computer worden noch spaties afgekapt noch door een vertaalprogramma spaties toegevoegd. In voorbeeld 6 wordt dit getoond.

#### 5.4 H e t   d o l l a r t e k e n

Het gebruik van het dollarteken (\$) als scheidingstekens tussen fortran-opdrachten op een fortranregel is bij een Vax-computer niet toegestaan. Elke fortranopdracht moet op een aparte fortranregel staan. Zie voorbeeld 7 waarin dit verschil tot uitings wordt gebracht.

Voorbeeld 7. Op de Vax geen dollartekens.

| Cyber                     | Vax   |
|---------------------------|-------|
| -----                     | ----- |
| K=0   \$   L=0   \$   M=0 | K=0   |
|                           | L=0   |
|                           | M=0   |

#### 5.5 H e t   s e l i j k t e k e n

Het herhaald doen voorkomen van het selijkteken (=) in een fortran-opdracht is in tegenstelling tot Cyber-fortran voor Vax-fortran niet toegestaan. Hier is slechts een selijkteken per fortranregel toegestaan. In voorbeeld 8 wordt dit verschil tussen Cyber-fortran en Vax-fortran weergegeven. De volgorde van ontwikkelen is hierbij van rechts naar links.

Voorbeeld 8. Op de Vax geen herhalingen van het selijkteken in een fortranregel.

| Cyber          | Vax   |
|----------------|-------|
| -----          | ----- |
| I=5   \$   J=2 | I=5   |
| K=L=I+J        | J=2   |
|                | L=I+J |
|                | K=L   |

## 5.6 Het CHARACTER datatype

Wanneer in een programma alphanumerieke tekens in een variabele worden ondergebracht, hetzij door deze variabele in te lezen via de A-specificatie van een FORMAT, hetzij door in deze variabele door middel van een toekenningsopdracht via een H-specificatie een keten van tekens onder te brengen, dan moeten deze variabelen vooraf worden gedefinieerd als zijnde van het CHARACTER datatype. In de toekenningsopdracht moet dan de H-specificatie worden weggelaten en de keten van alphanumerieke tekens moet worden omsloten door apostrophetekens. In voorbeeld 9 wordt dit verschil tot uitings gebracht.

Voorbeeld 9. Alphanumerieke tekens.

| Cyber              | Vax (eenvoudigst)     | Vax (aanbevolen)      |
|--------------------|-----------------------|-----------------------|
| -----              | -----                 | -----                 |
| REAL NAAM(20)      | CHARACTER*10 NAAM(20) | CHARACTER*10 NAAM(20) |
| .                  | .                     | CHARACTER*5 EIND      |
| .                  | .                     | .                     |
| .                  | .                     | .                     |
| DO 10 I=1,10       | DO 10 I=1,10          | DO 10 I=1,10          |
| READ(2,20) NAAM(I) | READ(2,20) NAAM(I)    | READ(2,20) NAAM(I)    |
| 10 CONTINUE        | 10 CONTINUE           | 10 CONTINUE           |
| 20 FORMAT(A10)     | 20 FORMAT(A10)        | 20 FORMAT(A10)        |
| EIND=5HEINDE       | EIND=5HEINDE          | EIND='EINDE'          |

## 5.7 Test op end-of-file

Bij Cyber-fortran komt een test op end-of-file direct na de leesopdracht in een aparte verselijkensoopdracht, daarentegen wordt bij Vax-fortran in de leesopdracht een parameter toegevoegd waarmee hetzelfde resultaat wordt beoogd. In voorbeeld 10 worden deze verschillen weergegeven.

Zoals uit voorbeeld 10 blijkt wordt in de leesopdracht de parameter END=11 toegevoegd. Het getal 11, dat vrij te kiezen is, is het opdracht-nummer waarnaar gesprongen wordt wanneer er een end-of-file wordt gelezen. Wanneer vanaf de terminal wordt gelezen, dan kan door het indrukken van de controoltoets tezamen met de Z-toets (/CTRL,Z/) een end-of-file worden gecreëerd. Op file kan slechts eenmaal een end-of-file voorkomen en wel aan het einde van de file.

Voorbeeld 10. Test op end-of-file.

| Cyber                            | Vax                              |
|----------------------------------|----------------------------------|
| -----                            | -----                            |
| CALL CONNEC(3)                   |                                  |
| DO 10 I=1,1000                   | DO 10 I=1,1000                   |
| READ(2,*) A(I),B(I)              | READ(2,*,END=11) A(I),R(I)       |
| IF(EOF(2).NE.0) GOTO 11          |                                  |
| 10 CONTINUE                      | 10 CONTINUE                      |
| 11 WRITE(3,100)                  | 11 WRITE(6,100)                  |
| 100 FORMAT(1H , 'INPUT GELEZEN') | 100 FORMAT(1H , 'INPUT GELEZEN') |

## 5.8 H u l p f u n c t i e s

De hulpfuncties welke zowel voor Cyber-fortran als Vax-fortran selden hebben dezelfde benamingen met uitzondering van functies welke betrekking hebben op intesers. Dit omdat een Vax twee verschillende inteser waarden kent n.l. INTEGER\*2 en INTEGER\*4. De intesers welke geen lengtezanduiding hebben, wat bij de conversie van Cyber naar Vax voor alle intesers geldt, zijn dan gedefinieerd als INTEGER\*4 en moeten daarom aan de functienaam de letter J als toevoeging hebben. In tabel 1 wordt een overzicht gegeven van alleen die hulpfuncties waarbij het resultaat of het argument van de hulpfunctie betrekking heeft op een inteser. Eventueel kan men op de Vax de in de eerste kolom genoemde functies als datatype voortbrengende naam gebruiken. Het datatype van het resultaat hangt dan af van het datatype van het argument (zie van DOORNE, 1982, pag 80 e.v.).

## 5.9 H u l p r o u t i n e s

Het gebruiken van de hulproutines heeft voor beide computers een verschillend karakter. Beide systemen kennen routines voor het bepalen van datum en tijd bijv:

| Cyber        | Vax               |
|--------------|-------------------|
| -----        | -----             |
| CALL TIME(a) | CALL TIME(a)      |
| CALL DATE(a) | CALL DATE(a)      |
|              | CALL IDATE(I,J,K) |

Tabel 1. Overzicht van hulpfuncties, waarbij het argument of resultaat betrekking heeft op een integer.

| FUNCTIE | TYPE VAN<br>ARGUMENT       | TYPE VAN HET<br>RESULTAAT  | CYBER                 | VAX                      |
|---------|----------------------------|----------------------------|-----------------------|--------------------------|
| ABS     | INTEGER                    | INTEGER                    | IABS                  | JABS                     |
| INT     | REAL<br>DOUBLE             | INTEGER<br>INTEGER         | INT<br>IDINT          | JINT<br>JIDINT           |
| MOD     | INTEGER                    | INTEGER                    | MOD                   | JMOD                     |
| MAX     | INTEGER<br>INTEGER<br>REAL | REAL<br>INTEGER<br>INTEGER | AMAXO<br>MAXO<br>MAX1 | AJMAXO<br>JMAXO<br>JMAX1 |
| MIN     | INTEGER<br>INTEGER<br>REAL | REAL<br>INTEGER<br>INTEGER | AMINO<br>MINO<br>MIN1 | AJMINO<br>JMINO<br>JMIN1 |
| FLOAT   | INTEGER                    | REAL                       | FLOAT                 | FLOATJ                   |
| FIX     | REAL                       | INTEGER                    | IFIX                  | JIFIX                    |
| SIGN    | INTEGER                    | INTEGER                    | ISIGN                 | JISIGN                   |
| DIM     | INTEGER                    | INTEGER                    | IDIM                  | JIDIM                    |

De aanroep hiervan is voor beide systemen gelijk, doch het datatype van het argument verschilt. Voor Vax-fortran geldt dat *a* van het CHARACTER datatype moet zijn (van DOORNE, 1982, pag 84). Dus :

voorbeeld 11. Het datatype van het argument voor het bepalen van datum en tijd.

```
CHARACTER*9 A  
CALL DATE(A)
```

De argumenten van de hulproutine CALL IDATE zijn van het integer datatype, dit houdt in dat de datum rekenkundig toepasbaar kan worden gemaakt en dat daarom de DECODE opdracht niet nodig is. Wil men daarentegen de tijd rekenkundig toepasbaar maken, dan zal wel gebruik gemaakt moeten worden van de DECODE opdracht.

Ook het verrichten van toevalstrekkings is voor beide systemen verschillend. Op een Cyber-computer ziet deze routine er als volgt uit:

voorbeeld 12. Het verrichten van toevalstrekkings.

```
Cyber:  CALL RANSET(X)  
        Y=RANF(I) , waarin  
  
        Y= het resultaat van de trekking  
        I= een dummy variabele
```

Voor een Vax-computer ziet deze routine er als volgt uit:

```
Vax:    Y=RAN(I) , waarin  
  
        Y= het resultaat van de trekking  
        I= een groot oneven integer getal.
```

De trekking van het toevalsgetal kan men laten afhangen van het tijdstip van de dag waarop het programma wordt gedraaid. Vooraf moet dan voor zowel de Cyber als de Vax computer een waarde bepaald worden voor respectievelijk de variabelen X en I. Dit kan worden gerealiseerd met behulp van de opdrachten:

voorbeeld 13. Het bepalen van een waarde afhankelijk van het tijdstip van de dag.

| Cyber                 | Vax           |
|-----------------------|---------------|
| -----                 | -----         |
| KLOCK=TIME(A)         | I1=SECNDS(.0) |
| DECODE(9,1,A)I1,I2,I3 | I =2*I1+1     |
| 1 FORMAT(3(1X,I2))    |               |
| X=3600*I1+60*I2+I3    |               |

## 6.0 SLOTPMERKING

In deze nota is slechts in het kort aangedeseven waar problemen zullen optreden bij het direct vertalen van Cyber-programma's op de Vax-computer en hoe dit kan worden voorkomen. Hier is dus een snelle manier beschreven om reeds bestaande programma's draaiklaar te maken voor de Vax. Hoe een fortran-iv programma om te zetten in een doelaatsig en overzichtelijk Vax-fortran programma is hier niet aangedeseven. De specifieke voordelen van Vax-fortran staan beschreven in ICW-nota 1393 (van Doorne, 1982).

Bijlage 2 laat zien hoe met behulp van de editor EDT aanhalingsstekens zijn te wijzigen in apostrophe's.

## REFERENTIES.

- DOORNE, W. van, 1982, Aspecten van Informatieverwerking 33. VAX-11 Fortran deel 1 (zonder input en output), ICW-nota 1393. 85 pp.
- OOSTINDIE, K. 1982, Aspecten van Informatieverwerking 34. Het interactief werken met een Vax-11 computer, ICW-nota 1382. 47 pp.
- MAASSEN, J.R. en K. OOSTINDIE, 1982, Aspecten van Informatieverwerking 34. Het gebruiken van de VAX-11 editor E D T, ICW-nota 1387. 57 pp.



Voorbeeld van de werking van programma TRAILING

\$ RUN [COIN.29990630]TRAILING

WAT IS UW DIRECTORY ?

[COIN.29990630]

HOE HEET DE FILE WELKE TRAILING BLANKS BEVAT?

FILE.TYP

HOE HEET DE NIEUW TE MAKEN FILE?

NIEUW.DAT

\*\*\*\*\* TRAILING BLANKS ZIJN VERWIJDERD \*\*\*\*\*

\$

BIJLAGE 2.

In deze bijlage wordt getoond hoe met behulp van de editor EDT alle aanhalingsstekens kunnen worden veranderd in apostrofes.

\* EDIT/EDT CYBER.FDR

1

PROGRAM ONE(INPUT,OUTPUT,TAPE1=INPUT,TAPE2=OUTPUT)

- S/'/'/'/' ZWH ! Alle enkele apostrophe's worden omgezet in dubbele
- S/'/'/' ZWH ! Alle aanhalingsstekens worden omgezet in apostrophe's

Eerst worden de reeds aanwezige enkele apostrophe's in twee apostrophe's omgezet, daarna worden alle aanhalingsstekens omgezet in apostrophe's. Het omzetten gebeurt met het SUBSTITUTE commando. Tussen scheidingstekens wordt eerst de te vervangen tekst opgegeven en daarna de vervangende tekst. Tenslotte wordt nog opgegeven op welk gedeelte van de file deze vervanging van toepassing is. In dit geval betreft het dus de gehele file (ZWHOLE). Zie ook ICW-nota 1387 (MAASSEN en OOSTINDIE,1982).