

Automatisering van de voorlichting inzake programmatuur

D. Bakker, U. Brouwer, P. Debets. *)

Samenvatting.

Een aantal argumenten worden gegeven voor het ter beschikking stellen van informatie m.b.t. het gebruik van programmatuur middels het computer-systeem waarop die programmatuur zelf te vinden is. Gepleit wordt voor een interactieve benadering.

Aangegeven wordt welke gegevens zo'n systeem idealiter zou moeten bevatten, en wat daarvan op dit moment gerealiseerd is in het programma TCINFO, zoals dat beschikbaar is op het Amsterdamse universitaire rekencentrum SARA. Tenslotte wordt enige aandacht besteed aan toekomstige uitbreidingen van het programma.

Key words: automatische programmadetektie, statistische programmatuur, methoden en technieken, question-answering systems, interactief-computergebruik.

0. Inleiding

In de afgelopen jaren is een aantal boeken verschenen, waarin problemen m.b.t. sociaal-wetenschappelijk onderzoek worden aangeduid (o.a. Soudijn, 1976; Hoogstraten, 1979 en van der Zee, 1979).

Voor al puur methodologische valkuilen en artefacten komen daarbij aan bod. Ook wordt ruim aandacht geschonken aan het interpreteren van onderzoeksresultaten.

Het is opvallend dat in geen van de genoemde werken wordt gerefereerd over de problemen die een onderzoeker tegen komt bij het, via de computer, verwerken van het onderzoeksmateriaal. Er bestaat een duidelijk niemandsland tussen de keuze van een analysetechniek en het interpreteren van onderzoeksresultaten.

In de praktijksituatie binnen dienstverlenende centra m.b.t. voorlichting aangaande programmatuurgebruik is gebleken, dat geen dan wel inefficiënte voorlichting op dit gebied soms tot desastreuze gevolgen leidt.

In dit artikel willen we ingaan op de rol die de computer kan spelen in het communicatieproces voorlichter-onderzoeker.

*) Technisch Centrum FSW, Roeterstraat 15
Universiteit van Amsterdam.

In paragraaf 1 wordt aangegeven in hoeverre automatisering van de voorlichting inzake programmatuurgebruik o.i. gewenst is.

Paragraaf 2 geeft aan wat in een dergelijk voorlichtingsprogramma idealiter zou moeten worden opgenomen; we geven tevens enkele al gerealiseerde voorbeelden.

Paragraaf 3 tenslotte geeft enkele meer technische aspecten van door ons in dit kader vervaardigde programmatuur.

1. Waarom de computer inschakelen?

Bij de universitaire rekencentra is nog steeds sprake van een duidelijke groei. Deze betreft het aantal - vaak incidentele en relatief onervaren - gebruikers; de aanwezige computersystemen; de interactieve aansluitingen; en de voor uiteenlopende bewerkingen op een toenemend aantal toepassingsgebieden beschikbare software.

De behoefte aan voorlichting van uiteenlopende aard is, in samenhang hiermee, navenant sterk gestegen.

Wij willen nagaan, op welke wijze het computersysteem zelf ingeschakeld kan worden bij het verstrekken van informatie aangaande het gebruik ervan. Wij zullen ons daarbij beperken tot het gebruik van sociaal-wetenschappelijk relevante computerprogrammatuur; wij menen echter, dat e.e.a. in principe overdraagbaar is naar computertoepassingen op andere vakgebieden. Ook zullen wij voornamelijk uitgaan van een interactieve gebruikssituatie: het laat zich aanzien dat, met het sneller worden van de processoren, en de groei van de geheugens, bij een toenemend aantal terminal-aansluitingen en het beschikbaar komen van steeds meer 'echte' interactieve programmatuur, deze wijze van computerverwerking de batch-verwerking in belangrijke mate zal gaan verdringen, althans voor wat betreft een substantieel deel van de nu bekende toepassingen.

Men kan zich afvragen welke voordelen zijn verbonden aan het inschakelen van de computer bij de voorlichting m.b.t. het gebruik van programmatuur, waar vaak voortreffelijke handboeken en programmabeschrijvingen voor handen zijn, en geschoolde en ervaren voorlichters van vlees-en-bloed tot onze dienst staan.

Voorlichting is een 'interactief' proces in optima forma, waarbij zich, afhankelijk van het kennisnivo van voorlichter en voorlichting vragende,

en de - meer of minder konkrete - problemen die aan de orde worden gesteld, tal van complexe en subtiële situaties kunnen voordoen.

De huidige stand van zaken op het gebied van de bij het 'opvoeden' van de computer betrokken wetenschaps-gebieden, waaronder de (cognitieve) psychologie, de informatika, en de linguïstiek annex de tekstwetenschap, al of niet opererend onder de paraplu van de kunstmatige intelligentie, geeft nog nauwelijks uitzicht op spoedige realisatie van een echte 'intelligente' dialoog tussen machine en mens over een voor de laatste interessant onderwerp. Gepaste bescheidenheid t.a.v. mogelijkheden en verwachtingen in dezen op kortere termijn lijkt dan ook geboden.

Dit neemt niet weg, dat er, zeker in het geval van een groot computersysteem met vele, op uiteenlopende lokaties actieve gebruikers, toch enige argumenten zijn voor het onderbrengen van bepaalde, op aanvaardbare wijze te presenteren gebruikers-informatie op dat systeem zelf.

Zo kan een computer-gestuurd informatiesysteem dat door één instantie wordt beheerd, beter compleet en up to date gehouden worden dan de gebruikelijke, op vele plaatsen aanwezige, en door uiteenlopende personen bij te houden dokumentatie-mappen. Het aantal menselijke handelingen dat gemoeid is met het onderhoud van zo'n systeem zal uiteindelijk aanmerkelijk geringer zijn, hetgeen de korrektheid van de beschikbare informatie eerder zal doen toe- dan afnemen.

Ook kunnen mutaties à la minute aan de gebruiker worden doorgegeven.

Een dergelijk systeem kan ook 'aktiever' zijn: het kan de gebruiker, eventueel afhankelijk van diens (gebleken) belangstellingssfeer, attenderen op de meest recente wijzigingen.

Vaak zijn eenvoudig voorzieningen aan te brengen die de gebruiker in de gelegenheid stellen voor hem interessante informatie te bewaren (hard copy); bij een schrijvende terminal gebeurt dit uiteraard reeds vanzelf. Bij een interactief systeem staan in principe twee kanalen open: de gebruiker kan immers de gelegenheid worden gegeven, opmerkingen, wensen, etc. ten aanzien van programmatuur, dokumentatie en andere zaken direkt aan de beherende instantie door te geven, al of niet met de uitvoer van geëxecuteerde jobs ter adstruktie.

Wordt door het systeem een boekhouding bijgehouden van de gevraagde informatie, dan kan de voorlichtende instantie op den duur een analyse maken van de aard van de informatie-behoefte, waaraan het systeem (en de rest van de voorlichtingsaktiviteiten) dan weer kunnen worden aangepast.

Een goed georganiseerd automatisch voorlichtingssysteem biedt i.h.a. meer ruimte voor een systematische en geïntegreerde opbouw van de diverse soorten informatie dan de, vaak over vele handboeken verspreide 'papieren' dokumentatie. Het kan beter dan - voor de 'algemene' c.q. 'gemiddelde' gebruiker samengestelde - dokumentatie op schrift, zó worden ingericht, dat het zich aanpast aan het kennisnivo en de belangstelling van de gebruiker.

Het lijkt verder aantrekkelijk, als bepaalde informatie verkregen kan worden via hetzelfde medium als waarmee men zijn 'eigenlijke' opdrachten ten uitvoer brengt. Voorts kunnen op deze wijze meerdere vragen om informatie tegelijkertijd worden afgehandeld, en ontstaat er geen wacht-rij.

De gebruikelijke voorlichtingsbalies zijn meestal gedurende een (veel) geringer aantal uren opengesteld en bemand dan het computersysteem 'op' is, zodat men buiten de normale uren vaak nauwelijks aan informatie kan komen, ook niet telefonisch. Bovendien bevinden de balies zich lang niet altijd op loopafstand van de plaatsen waar terminals staan opgesteld. Slaagt men erin de stroom van meer eenvoudige en routinematige vragen op bevredigende wijze te verleggen naar het computersysteem, dan betekent dit ongetwijfeld een ontlasting voor de voorlichtende instanties, die zich dan in ruimere mate met de diepergravende en interessantere problemen (de tweede-lijnsvoorlichting) kunnen gaan bezighouden.

Tenslotte kan men stellen, dat de loutere aanwezigheid van een geautomatiseerd voorlichtingssysteem de beheerders ervan zal nopen tot een grotere mate van explicitering, systematisering en een beter onderhoud t.a.v. de aan de gebruikers beschikbaar te stellen informatie, dan onder de gebruikelijke omstandigheden, ook bij de beste bedoelingen, vaak het geval is.

2. Welke informatie te automatiseren?

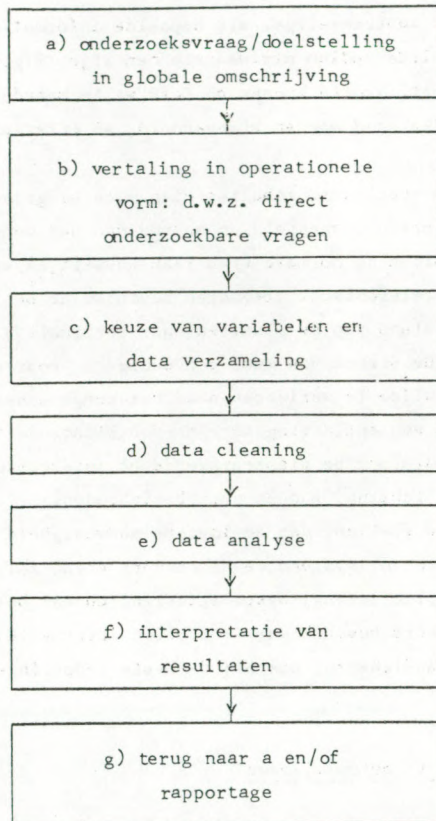
We willen nu ingaan op de vraag, welke informatie zich bij uitstek leent voor opneming in een automatisch voorlichtingssysteem. We doen dit aan de hand van voorbeelden ontleend aan het programma TCINFO (Bakker et.al. 1980), een programma dat beoogt in (een deel van) deze behoefte te voorzien.

Achtereenvolgens stellen we aan de orde:

- 1) wat zou idealiter in een informatiesysteem dienen te worden opgenomen,
- 2) wat is hiervan op dit moment gerealiseerd in TCINFO,
- 3) welke uitbreidingen zijn voorzien.

2.1 Wat bevat een ideaal informatie-systeem m.b.t. programmatuur-voorlichting?

Teneinde deze vraag te kunnen beantwoorden, willen we eerst eens nagaan wat bij een mondelinge interactie tussen de konsulent en de cliënt zoal aan de orde komt. We willen dit toelichten aan de hand van onderstaande afbeelding.



Afbeelding 1: mogelijke volgorde van onderzoeksfasen.

Het spreekt voor zich dat een centrum dat computerprogrammatuur ter beschikking stelt aan derden, informatie geeft over die programmatuur en de keuze

om tot optimale verwerkingsprocedures te komen helpt vergemakkelijken. De konsulent die voorlichting geeft inzake de computer-verwerkingsfase heeft in principe dan ook slechts te maken met fasen d en e uit de voorgaande afbeelding.

In de praktijk komt het echter voor dat een "onderzoeker" met een bak ponskaarten aan de voorlichtingsbalie verschijnt met de vraag: hoe moet dit de computer in, wat moet ik hiermee doen, wat voor vragen moet ik hiermee beantwoorden, en meer van dergelijke curiosa. Ook wordt regelmatig computeruitvoer op de balie gedeponeerd met het verzoek uitleg te geven. De "onderzoeker" is dan niet in staat zelf de gevonden resultaten te interpreteren. Wanneer we deze uitwassen buiten beschouwing laten, dan hebben we doorgaans met de volgende vragen te maken:

- 1) Is er een methode om ... (volgt globale probleemstelling).
- 2) Bestaat er een programma voor
(en dan wordt een statistische methode globaal of soms ook zeer gedetailleerd omschreven).
- 3) Wat is er eigenlijk fout aan het programma XXX?
- 4) Waar kan ik literatuur vinden over methode X (in programma YYY)?
- 5) Is programma 1 voor faktoranalyse beter dan programma 2?
- 6) Wat heb ik fout gedaan in de setup van deze computer run? (een listing wordt hierbij ter inzage gegeven)
- 7) Hoe moet ik mijn probleem vertalen in een setup voor een computer run?
- 8) Mag ik eens een voorbeeld uitvoer van programma XXX zien?
- 9) Vragen m.b.t. het besturingssysteem van de computerinstallatie (reken-tijd, geheugenbeslag, benodigde programmabibliotheken, enz.).

De 9 soorten vragen hier opgesomd vormen meer dan 80% van het totale aantal dat bij een voorlichtende instantie als het Technisch Centrum FSW binnenkomt.

Deze vragen lenen zich slechts ten dele voor beantwoording via geautomatiseerde voorlichting.

Om ze in hun volle omvang aan te kunnen, dienen we met name te beschikken over een geavanceerd nivo van kennisopslag in de computer m.b.t. het probleemgebied, en routines die in staat zijn door de gebruiker te formuleren problemen te analyseren en te interpreteren.

Het ziet er naar uit dat op deze beide gebieden in de nabije toekomst grote vooruitgang kan worden geboekt. Op dit moment is het zeer wel denkbaar

in elk geval allerlei aspecten verbonden aan de 9 genoemde vraagtypen in een systeem onder te brengen.

Dit zou dan moeten bevatten:

- a) Algemene informatie over alle beschikbare programma's binnen een rekencentrum. Hieronder verstaan we:
 - 1) korte omschrijving van de programma's
 - 2) verwijzing naar gebruikersdokumentatie
 - 3) verwijzing naar literatuur die relevant is m.b.t. de geprogrammeerde methode.

Het moet mogelijk zijn dat de gebruiker zijn vragen doorgeeft aan het systeem, en na een aantal keuzes gemaakt te hebben, uiteindelijk een gewenst computerprogramma vindt, waarbij dan tevens de 3 bovenstaande punten beantwoord zijn.

- b) Meer technische informatie over elk programma. Dit zou kunnen inhouden:

- 1) Informatie over het programma in relatie met de beschikbare computerinstallatie b.v. geheugenbeslag, gemiddelde rekentijd, hoe externe files binnen te halen, gebruik van bibliotheken, hoe is het programma aanroepbaar, enz.
- 2) Het presenteren van een "standaard" of doorsnee setup met bijbehorende uitvoer. De uitvoer moet dan uitvoerig geannoteerd zijn.
Dit geeft de gebruiker enig inzicht in de gebruikersvriendelijkheid van het verlangde programma: Valt er makkelijk of lastig mee te werken?

- c) Gedetailleerde informatie over elk programma.

- 1) Een overzicht, liefst in schematische vorm, van de geïmplementeerde methoden en technieken inclusief de verschillende opties voor invoer en uitvoer.
- 2) Een lijst met bekende fouten in de programma's. Hierbij dient tevens te worden opgenomen, wanneer de fout is ontdekt en wanneer deze wordt opgelost. Aanwijzingen moeten worden gegeven hoe het gerezen probleem te omzeilen, b.v. door het aangeven van vergelijkbare programma's, waarmee hetzelfde gewenste resultaat is te behalen.

- d) Informatie m.b.t. de vergelijkbaarheid van programma's.

Er bestaan b.v. tientallen faktoranalytische methoden met bijbehorende programmatuur. Voor een gebruiker is het uiterst lastig een optimale

keuze te maken uit deze programma's. Via het opnemen van evaluatie-tabellen in het informatiesysteem kan de gebruiker zelf "vergelijkbare" programma's op hun merites beoordelen. Deze informatie vergemakkelijkt het maken van een optimale keuze.

Tot nu toe is uitsluitend ingegaan op informatie verbonden aan één bepaald programma (a, b, c) of aan een aantal vergelijkbare programma's (d).

We achten het verder wenselijk ook meer algemene informatie m.b.t. bepaalde statistische methoden en technieken in het informatiesysteem op te nemen. Hierbij denken we primair aan:

e) Het toetsen van vooronderstellingen.

Tot op heden bevat vrijwel geen enkel statistisch programma of pakket de mogelijkheid om vooraf na te gaan of aan de assumpties voor een bepaalde techniek wordt voldaan. De meest "zieke" datasets losgelaten op een statistisch programma, leveren wel (interpreteerbare) output op. Bij elke analysetechniek zou op verzoek van de gebruiker moeten worden aangegeven welke vooronderstellingen er voor de betreffende techniek gelden. De onderzoeker zou z'n dataset moeten kunnen invoeren, om via de "vooronderstellingen" module van het informatieprogramma te kunnen toetsen hoe goed zijn data aan de vereiste assumpties voldoen.

Het lijkt zinvol per methode en/of techniek aan te geven welke assumpties er veel of relatief weinig toe doen, m.a.w. hoe robuust de betreffende methode is met betrekking tot inbreuken op bepaalde vooronderstellingen.

Vaak komt het voor dat de onderzoeker na het verzamelen van onderzoeksmateriaal nog weinig inzicht heeft in de aard van die gegevens. Alhoewel hij doorgaans weet welke statistische analyse hij wil uitvoeren, is het toch raadzaam vooraf eens naar de data te kijken. En dat ook in de letterlijke betekenis, m.a.w. we achten het zinvol dat een informatiesysteem, zoals we dat hier beschrijven, ook de mogelijkheid bevat de verzamelde gegevens te bekijken:

f) Inspectie van data bestanden.

We rekenen hiertoe o.a.:

- eenvoudige plot-uitvoer van de data
- berekenen van eenvoudige descriptieve grootheden, zoals gemiddelden, range, varianties en correlaties
- detektie van uitschieters

- inspectie van berekende correlatie-matrices (shading, faktoriseerbaarheid, enz.)

Het spreekt vanzelf dat bij mogelijkheid e en f de data door het informatiesysteem worden ingelezen. Het lijkt ons uiterst zinvol alvorens een "echte" analyse uit te voeren te onderzoeken, in hoeverre een geplande analyse ook werkelijk mag worden toegepast.

We zijn ons ervan bewust dat het op korte termijn niet mogelijk is alle hierbij benodigde informatie in één hanteerbaar programma op te slaan. Evaluatiegegevens zijn b.v. slechts na tijdrovende studies te verzamelen. Ook vergt het opzetten en uitwerken van een programma (of module) voor inspectie van data bestanden, inclusief toetsen op vooronderstellingen, veel onderzoek en mankracht.

Wat is nu eigenlijk wel en wat niet op overzienbare termijn te implementeren? Welke toetsen zijn makkelijk te algoritmiseren en welke erg moeilijk? Hoe kunnen we gebruik maken van exploratieve data-analyse (EDA) technieken?

Al dit soort vragen moeten in de nabije toekomst worden beantwoord. Dit neemt echter niet weg dat op dit moment een informatiesysteem op het Technisch Centrum beschikbaar is, dat een aantal van de boven geschetste informatiebronnen bevat.

2.2 Wat bevat TCINFO op dit moment?

TCINFO geeft informatie over de bij het Technisch Centrum in beheer zijnde programma's op het gebied van:

- Statistiek
- Data manipulatie
- Data cleaning
- Information retrieval
- Tekstverwerking
- Gedragsanalyse
- Algemene utilities
- Edukatieve programma's
- Spelprogramma's

Het grootste deel van de (+ 300) programma's zijn statistisch van aard (ongeveer 65 %).

Over deze programma's geeft TCINFO de volgende informatie:

- een omschrijving van het programma
 - de vindplaats
 - bijzonderheden van het programma
 - verwijzing naar dokumentatie,
- en bovendien informatie over "bekende" fouten in de programma's:
- een omschrijving van de fout
 - datum wanneer de fout gemeld is en/of opgelost,
 - een mogelijk alternatief.

De meest eenvoudige manier om informatie in TCINFO op te vragen is via de programma-namen. Een voorbeeld hiervan:

WILT U INFORMATIE OVER EEN U REEDS BEKEND PROGRAMMA ?

ja

TYPE PROGRAMMA NAAM OF EERSTE LETTER(S) GEVOLGD DOOR +

multivariate

MULTIVARIANCE

PROGRAMMA MULTIVARIANCE

PAKKET TCLIB

OMSCHRIJVING MULTIVARIATE (CO) VARIANTIE-ANALYSE. FIXED EN RANDOM MODEL. CROSSED EN NESTED DESIGNS MET OF ZONDER COVARIATEN. OOK NIET-ORTHOGONALE DESIGNS. VERSCHILLENDE SOORTEN CONTRASTEN EN POLYNOMEN MOGELIJK.

VINDPLTS TCLIB. PROGRAMMA NAAM : MULTIVA.

BIJZON- LIBRARY : TCLIB.

DOKUM- MULTIVARIANCE USER'S GUIDE VERSION VI , RELEASE 2

A T T E N T I E : ER ZIJN 2 FOUTEN BEKEND; WILT U DEZE ZIEN ?

ja

MULTIVARIANCE

PROGRAMMA MULTIVARIANCE

PAKKET TCLIB

FOUT IN SOMMIGE OMSTANDIGHEDEN KAN MULTIVA I.P.V. OUTPUT DE MELDING "MODE ERROR" GEVEN. OORZAAK IS VERMOEDELIJK EEN "REDUCE" IN DE SETUP.

MULTIVARIANCE

PROGRAMMA MULTIVARIANCE

PAKKET TCLIB

FOUT ALS EEN AFHANKELIJKE VARIABLE INGELEZEN WORDT MET I-FORMAAT (MAG NIET, BLZ. 75) DAN GEEFT MULTIVA GEEN MELDING MAAR GAAT LUSTIG REKENEN, WAARBIJ ALLE CELLEN NUL WORDEN.

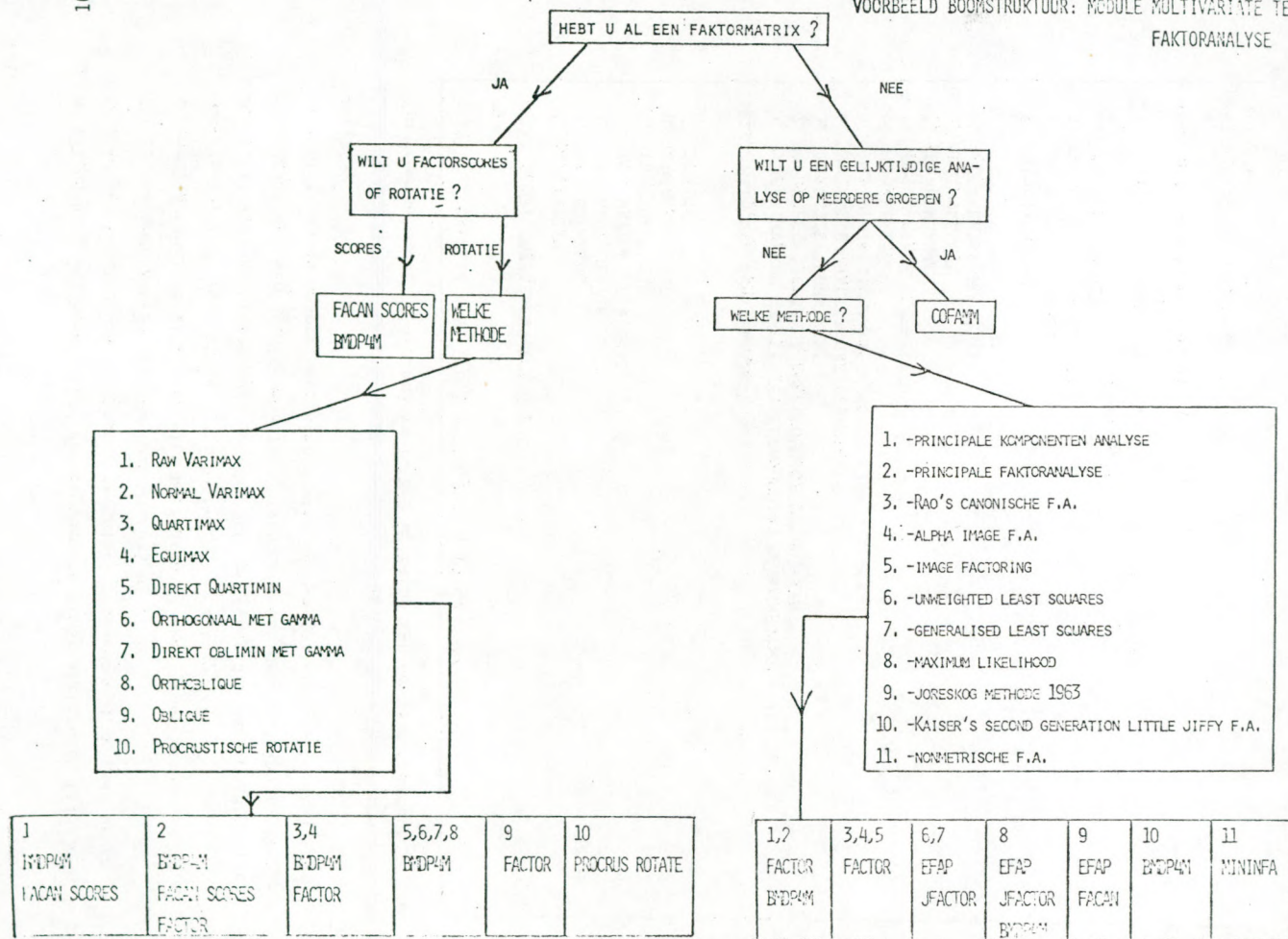
DATUM MLD 4-9-81

WILT U DEZE INFORMATIE BEWAREN OP DE FILE OUTPUT ?

nee

afb. 2: voorbeeld informatieblok.

Meestal weet een onderzoeker welke techniek hij wil gebruiken, maar kent hij geen bij die techniek behorend programma. TCINFO biedt dan de mogelijkheid interactief te zoeken naar een programma door een reeks vragen af te werken. Hiertoe zijn de statistische programma's onderverdeeld in een aantal categorieën en wordt binnen elke categorie een boomstructuur afgelopen. Een voorbeeld van zo'n hiërarchische structuur is onderstaand schema voor de techniek faktoranalyse.



We willen hier wel benadrukken dat de boomstructuur zeker niet bedoeld is voor het bepalen van de "beste" techniek. Integendeel: we gaan ervan uit dat de onderzoeker precies weet wat hij wil.

In sommige gevallen wijst het niet begrijpen van een vraag erop, dat de onderzoeker niet goed thuis is in de betreffende techniek.

D.m.v. literatuurverwijzingen of zgn. HELP routines die de vraagstelling verduidelijken kan TCINFO voorzien in een deel van deze problemen. Meestal echter zal de onderzoeker zich moeten wenden tot een statistisch consultant.

Een voorbeeld van de dialoog in TCINFO op basis van de boomstructuur:

WILT U INFORMATIE OVER EEN U REEDS BEKEND PROGRAMMA?

nee

WELKE SOORT TOEPASSING ZOEKT U ?

1. STATISTIEK
2. DATA CLEANING
3. DATA MANIPULATIE
4. INFORMATION RETRIEVAL
5. TEKSTVERWERKING
6. GEDRAGSANALYSE
7. OVERIGE

1

WELKE METHODE/TECHNIEK Kiest U ?

1. BESCHRIJVENDE STATISTIEK (EENVOUDIGE GROOTHEDEN EN ASS.MATEN)
2. HYPOTHESE TOETSENDE ENKELVOUDIGE TECHNIEKEN
3. REGRESSIE TECHNIEKEN
4. VARIANTIE-ANALYTISCHE TECHNIEKEN
5. MULTIDIMENS.SCHAALTECHNIEKEN (INKL. 1 DIM.)
6. TIJDREEKSEN
7. MULTIVARIATE TECHNIEKEN
8. GRAPHEN ANALYSE
9. TESTMODELLEN
10. LOGLINEAIRE MODELLEN

7

1. FACTORANALYSE
2. DISCRIMINANT ANALYSE
3. CANONISCHE CORRELATIES
4. CLUSTER ANALYSE
5. LINEAIRE STRUCTURELE MODELLEN
6. TOETS OF PATROONHYPOTHESEN IN CORRELATIE MATRICES

1

HEBT U AL EEN FACTOR MATRIX ?

ja

WILT U FACTOR SCORES OF ROTATIE ?

rotatie

1. RAW VARIMAX
 2. NORMAL VARIMAX
 3. QUARTIMAX
 4. EQUIMAX
 5. DIRECT QUARTIMAX
 6. ORTHOGONAAL MET GAMMA
 7. DIRECT OBLIMIN. MET GAMMA
 8. ORTHOBLIQUE
 9. OBLIQUE
 10. PROCRUSTISCHE ROTATIE
- 2

U HEEFT DE KEUZE UIT DE VOLGENDE PROGRAMMA'S :

- 1 BMDP 4M
- 2 FACAN SCORES
- 3 FACTOR

U WILT MEER INFO M.B.T. NUMMER ... ? (0 = GEEN)

2

FACAN SCORES

PROGRAMMA FACAN SCORES

PAKKET STAP

OMSCHRIJVING FACTOR SCORES VOLGENS DE ONDERSTAANDE 3 METHODEN

- 1) REGRESSIE
- 2) BARTLETT
- 3) GEGENERALISEERDE ANDERSON-RUBIN

afb. 3: voorbeeld dialoog

Nadat de voor een bepaalde techniek beschikbare programma's gegeven zijn, bestaat de mogelijkheid tot het opvragen van de algemene en fouten informatie over die programma's.

2.3 Toekomstige plannen.

In 2.1. is ingegaan op de ideale inhoud voor een interactief informatie-systeem. Paragraaf 2.2 laat zien wat wij hiervan tot nu toe verwezenlijkt hebben. In de nabije toekomst zal TCINFO uitgebreid worden met de volgende informatiebronnen:

1. Informatie over literatuur.

Op korte termijn hopen we een deel van de boeken uit de bibliotheek van het Technisch Centrum voorzien te hebben van trefwoorden die betrekking hebben op sociaal-wetenschappelijke computertoepassingen. Deze informatie zal dan via TCINFO beschikbaar zijn en de gebruiker kan op die manier zoeken naar literatuur over een bepaald onderwerp. In beperkte mate is dit in de huidige versie van TCINFO reeds mogelijk, echter het aantal trefwoorden en het aantal ingevoerde boektitels is nog gering.

2. Evaluatiegegevens.

Om een keuze te kunnen maken zal de gebruiker de beschikking moeten hebben over tabellen, waarin programma's geëvalueerd, en onderling vergeleken worden.

Daarbij komen de volgende punten aan de orde (zie ook Brouwer en Debets, 1980):

- a. Wat zijn de mogelijkheden van elk programma?
Overzichten van invoeropties, methoden en uitvoeropties.
- b. Hoe ziet de gebruikersdokumentatie eruit?
Bevat de manual: - een methodologische inleiding,
- een beschrijving van alle invoeropties,
- niet meer dan het programma aankan,
- voorbeeld in- en uitvoer.
- c. Hoe gebruikersvriendelijk is een programma?
Dit is o.m. afhankelijk van:
 - de dokumentatie,
 - stuurtaal (o.m. fixed format of keywords),
 - fouten afhandeling,
 - weergave van de invoer,
 - leesbaarheid van de uitvoer.
- d. Hoe doen de programma's het in de praktijk?
- e. Worden modelassumpties vooraf getoetst?
- f. Hoe worden ontbrekende gegevens verwerkt?

Met name de punten a, b en c kunnen in tabelvorm aan de gebruiker gepresenteerd worden als een soort consumentenoverzicht, aan de hand waarvan hij voor zichzelf een oordeel kan vormen. Onderstaande tabel is een voorbeeld van een vergelijking van de programma's voor faktoranalyse.

	ruwe data	correlatie matrix	covariantie matrix	factor- ladingen	factorkore coëfficiënten
BMDP4M	x	x	x	x	x
EFAP	x	x	x		
FACAN	x	x	x		
FACTOR	x	x		x	
JFACTOR	x	x			

afb. 4: Invoer mogelijkheden van de programma's.

3. Toetsen van vooronderstellingen.

Het woord 'toetsen' is hier misschien iets te weids: de meeste multivariate technieken hebben vooronderstellingen die niet dan wel moeilijk toetsbaar zijn. Inspectie van de data is echter relatief eenvoudig uitvoerbaar. Voorlopig zullen enkele mogelijkheden voor het uni-, bi- en multivariate geval ondergebracht worden in een apart programma.

4. Voorbeeld in- en uitvoer.

TCINFO zal op verzoek voorbeelden van de werking van het programma laten zien.

Meestal zullen deze voorbeelden wel voorzien moeten zijn van uitgebreid commentaar.

Iets verder in de toekomst gaan onze gedachten uit naar uitbreiding van TCINFO met:

- "HELP-routines" die vragen verduidelijken, maar ook uitleggen wat bijvoorbeeld faktoranalyse is. Probleem hierbij is dat er moeilijk een grens te trekken is tot hoever men hier moet gaan: TCINFO is niet als edukatief systeem bedoeld.
- "HELP-routines" die helpen bij het opbouwen van de setup voor een programma. Dit kan in de vorm van een soort interface waarin d.m.v. vragen de stuurkaarten opgebouwd worden (denk aan CLUSCOM, waarmee men interactief een setup opbouwd voor het CLUSTAN pakket) of in de vorm van het laten zien van het algemeen formaat van een bepaalde stuurkaart.

We gaven een schets van de voorziene uitbreidingen in het voorlichtingssysteem; hoge prioriteit zal verder worden gegeven aan opmerkingen en verlangens van de gebruikers.

3. Het programma

In het voorafgaande refereerden we enige malen aan het programma TCINFO zoals dat reeds enkele jaren beschikbaar is op het Amsterdamse Universitaire rekencentrum SARA. We zullen hier wat nader ingaan op de meer technische kanten van het programma, en op een geheel nieuwe versie, die momenteel op stapel staat.

Wellicht de belangrijkste aspecten van een op de computer geïmplementeerd vraag-antwoord systeem zijn:

- de aard van de informatie die erin is opgeslagen,
- de organisatie van die informatie, en
- de wijze waarop met het programma kan worden gekonverseerd.

In het kort: wat het programma 'weet' en hoe daar achter te komen.

Bij de huidige versie van het programma behoren drie gegevensverzamelingen: een random file met programmabeschrijvingen, een met foutenbeschrijvingen en een met boektitelbeschrijvingen.

Aan het begin van een sessie selekteert de gebruiker hetzij de eerste twee, hetzij de laatste file. Tevens laadt het programma van schijf een (gealfabetiseerde) index op de respektieve bestanden, die snel zoeken mogelijk maakt. Geheugenbeperkingen sluiten onder de huidige omstandigheden uit, dat alle drie files met hun indices tegelijkertijd ter beschikking staan, zodat informatieverschaffing over programma's enerzijds, en literatuur anderzijds, in aparte sessies plaats moet vinden.

Naast deze, verder ongestructureerde files, bestaat de 'kennis' van het programma over het 'onderwerp-van-gesprek' verder uit een hiërarchische opdeling van computertoepassingen en sociaal-wetenschappelijke methoden en technieken in de vorm van de boomstructuur waarvan wij eerder fragmenten toonden. Deze is niets anders dan een geordende verzameling vragen, als vaste tekst in het programma opgenomen, die achtereenvolgens beantwoord dienen te worden, waarna uiteindelijk een of meer programmanamen op de terminal verschijnen. Via deze namen kan informatie uit de beide programmafiles worden geselecteerd (dat laatste kan overigens ook direkt via de programmanamen, dus zonder voorafgaand vraag-antwoord spel, of, indirect, via de naam van een pakket of programmabibliotheek).

Deze informatie wordt desgewenst vastgehouden, zodat na afloop een afdruk op papier kan worden vervaardigd.

In de literatuur-module worden de gebruiker telkens vier vragen voorgelegd (auteursna(a)m(en); titel; jaar of periode; trefwoord) die hij dan al of niet invult. Op basis van deze sleutels worden vervolgens titels geselecteerd.

Er zijn, gemeten naar het type antwoord, drie soorten vragen:

Allereerst de 'ja-nee' vraag:

(1) wilt u rotatie?

De gebruiker antwoordt met j(a) of n(ee).

Daarnaast zijn er vragen met twee of meer alternatieven, die in de volgende vormen voorkomen:

(2) univariaat of multivariaat?

(3) kiest u uit de volgende mogelijkheden:

1. statistische bewerkingen
2. data cleaning
3. data manipulatie
4. tekstverwerking
5. overige

Bij (2) typt men het relevante begrip (men kan volstaan met de meest significante letter(s)) bij (3) typt men het cijfer behorende bij het gekozen alternatief.

Tenslotte zijn er de open vragen:

(4) typt u de naam van een programma, a.u.b.

Het getypte antwoord wordt dan in zijn totaliteit opgevat als zoek sleutel. Na een verkeerde selectie kan men, via het typen van een asterisk (*) terugkeren naar de vorige vraag.

Bij de vragen over literatuur kan men zijn vorige antwoord op de bewuste vraag continueren door het typen van een gelijkeken (=). Dit is met name praktisch als men in een vorige vragenronde te weinig specifiek is geweest, en diensgevolge te veel titels heeft geselecteerd. Men kan dan door eerst opengelaten vragen nu wel in te vullen een fijnere selectie maken.

Verder is het bij elke vraag mogelijk, door het typen van een vraagteken (?) enige uitleg te verkrijgen. In veel gevallen is dit in de huidige versie overigens niet meer dan een overzicht van de verwachte antwoorden.

Een laatste aspekt op het kommunikatieve nivo is enige soepelheid waar het om de acceptatie van ingetypte antwoorden gaat. Typt men een niet-bestaande programmaam, dan zoekt het programma in de index de meest gelijkende namen en legt deze aan de gebruiker voor. Bij de boektitels worden, in geval geen sprake is van exacte gelijkheid met een of meer bekende titels, al die titels geselecteerd, die een of meer woorden bevatten die (ongeveer) overeenkomen met de gezochte titel. Daarbij worden o.m. lidwoorden en voorzetsels buiten beschouwing gelaten.

Aan het eind van een sessie kan de gebruiker problemen, wensen en opmerkingen kenbaar maken, welke op een aparte file worden opgeslagen, en periodiek bekijken; 'terugkoppeling' vindt zondig plaats via de (interne) post.

Alhoewel het programma in een aantal opzichten bevredigend werkt, is toch inmiddels een aantal bezwaren aan het licht getreden, die te maken hebben met de aan het begin van deze paragraaf genoemde hoofdpunten:

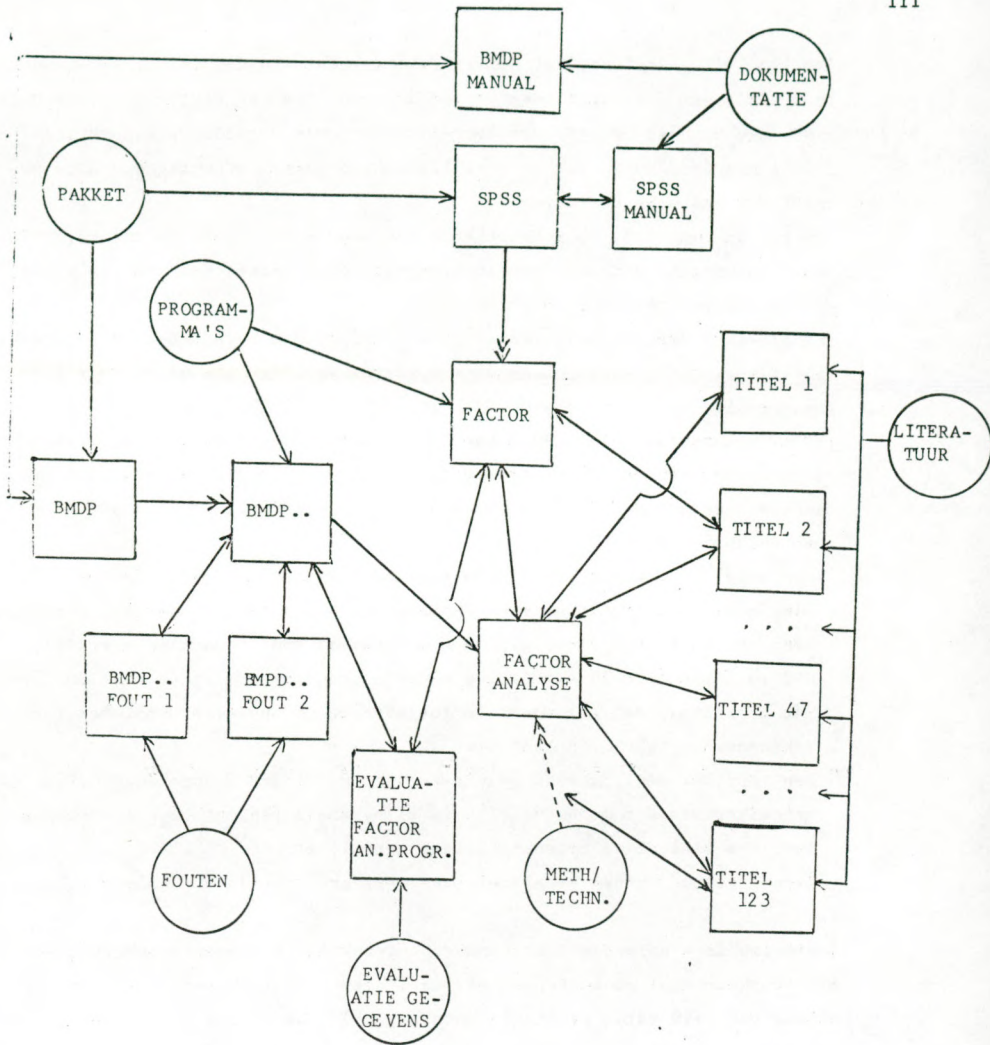
- er moet, bij de aanvang van de sessie, een keuze worden gemaakt tussen beschikbare data bases (programma informatie; literatuur). Geïntegreerde benadering van de gegevens door de gebruiker (van literatuur naar programma's; en vice versa) is niet mogelijk. Dit gaat te meer knellen bij de in de paragraaf 2 geschetste uitbreidingen (o.m. programma evaluatie gegevens).
- de 'dialoog' wordt volledig door het programma gestuurd.
Dit veronderstelt een opbouw van de vragen die voor alle gebruikers doorzichtig en eenduidig is. Met name voor de opdeling van de methoden en technieken lijkt dit illusoir, in elk geval discutabel.

In een op stapel staande, geheel nieuwe versie van het programma hopen wij o.m. via de hieronder aangeduide modificaties tegemoet te komen aan deze problematiek.

Allereerst de gegevensverzameling.

De geschetste spreiding van de informatie over aparte files maakt een geïntegreerde benadering dus goeddeels onmogelijk. Een adequaat informatiesysteem dient de beoogde koppeling (tussen programma's; hun fouten; de documentatie en literatuur; evaluatiegegevens; methoden en technieken) echter zeker tot stand te brengen.

Dit betekent dat we ernaar moeten streven, alle 'externe' informatie in één grote data base beschikbaar te hebben. Om het zoeken daarin te vergemakkelijken, en daarmee de respons-tijden acceptabel te houden, lijkt het aan te bevelen, de lineaire ordening van de records te verlaten, de samenhang die er is tussen de respektievelijke gegevens-eenheden expliciet tot uitdrukking te brengen, en het geheel de vorm te geven van een netwerk. Wij stellen ons zo'n data base als volgt voor:



afb. 5: data base organisatie

De cirkels zijn de 'ingangen' die verwijzen naar de betreffende categorieën van gegevens/records; de rechthoeken zijn de eigenlijke informatie dragende records.

Er zijn drie typen relaties, gerepresenteerd door drie soorten pijlen: de enkelpuntige pijl kan men lezen als: 'is van het type', 'is een ..'; de dubbel-gepunte pijl duidt op een subcategorie c.q. onderdeel; de aan beide zijden gepunte pijl legt een gelijkwaardige relatie.

Men kan, als gebruiker, het netwerk binnenkomen via een der cirkels, en er middels een dialoog verder in doordringen. Men kan echter ook, via het trefwoord van een record (een programmaam; een methode; een boektitel) direkt naar midden in het netwerk liggende informatie springen, en de betreffende gegevens opvragen.

Voorts kan men snel, door associatie van begrippen, naar een andere, relevante informatie-eenheid 'overstappen' via de diverse typen van relaties die de informatierecords verbinden.

'Kennis-van-zaken', zoals vervat in een dergelijke data base, is een noodzakelijke, echter geen voldoende voorwaarde voor een geslaagde voorlichtingssessie.

Het programma dient daarnaast een zeker 'idee' te hebben van het kennisniveau en de wensen van de gebruiker waarmee de interactie plaatsvindt. Het dient daartoe over een model te beschikken van de gebruiker, dat bestaat uit twee delen:

- een vast deel, waarin o.m. zijn opgenomen een aantal deduktieregels zoals: wie informatie vraagt over een programma, wil naast de algemene beschrijving ook de fouten zien; wie 'een programma' voor bewerking x vraagt, wil in feite álle in aanmerking komende programma's, en daarvan dan weer dát programma, dat het meest is toegesneden op bepaalde problemen (de rekennauwkeurigheid, opties etc.).
- een variabel deel, waarin gaandeweg informatie wordt opgebouwd m.b.t. de gebruiker van het moment: welke is zijn kennis van methoden en technieken; hoe ziet de te bewerken dataset eruit; etc.

Deze gegevens kunnen eventueel over meerdere sessies worden vastgehouden.

De variabele kennis dient tijdens de interactie te worden opgebouwd, en bij voorkeur niet op basis van een expliciete vraag ("geeft u aan op een schaal van 1-10 wat u kennis is inzake ..."). E.e.a. zal dus moeten worden gestedilleerd uit de informatie die de gebruiker - goeddeels onbewust - geeft over zichzelf tijdens de vraag-antwoord dialoog, zoals dat ook gaat in de normale voorlichtingssituatie.

Daartoe echter is de gedemonstreerde methode van de vastliggende vraag-boomstructuur - die ook al eerder genoemde bezwaren met zich meebracht - niet toereikend: uit de keuze immers van een categorie blijkt nimmer, of deze bewust en weloverwogen geschiedde.

Het programma dient dus op gezette tijden de gebruiker in de gelegenheid te stellen, zelf vragen en problemen te formuleren. Voor dit doel zijn

een aantal zgn. query-talen ontwikkeld, die echter als bezwaar hebben, dat men ze eerst moet leren, hetgeen een drempel betekent vooral voor de meer incidentele gebruiker van zo'n systeem, voor wie met name ons programma bedoeld is.

Bovendien is het de vraag, of men een dergelijke formele taal zò kan konstrueren, dat hij ook bij de beoogde uitbreidingen van de data base en de introductie van het gebruikersmodel nog kan dienen als adequaat communicatiemiddel.

Echter, een taal die de gewenste eigenschappen stellig bezit, en die tevens elke gebruiker reeds kent, is voor handen: de eigen taal, in ons geval het Nederlands. In tegenstelling tot het gebruik van een formele taal als intermediair, brengt het hanteren van een natuurlijke taal als belangrijkste probleem echter mee, dat geen enkele van deze talen, ook niet het Engels, volledig is beschreven, en in regels gevat. Het is zelfs de vraag, of dit ooit helemaal zal lukken. We zouden echter voorlopig kunnen volstaan met een voor de toepassing relevante en voldoende selectie.

Zo kunnen we, in ons geval, het vokabularium, naast funktiewoorden als lidwoorden en voorzetsels, die nu eenmaal altijd nodig zijn, beperken tot de terminologie van het (sociaal-wetenschappelijk) computergebruik. We kunnen beperkingen stellen aan het zinstype (alleen vragen), de complexiteit (geen bijzinnen), enzovoort.

We kunnen in het bestek van dit artikel niet uitgebreid ingaan op de problemen van de (automatische) interpretatie van Nederlandse en in andere natuurlijke talen gestelde zinnen en teksten.

Daarvoor verwijzen we naar de literatuur (w.o. Lehnert, 1978; Schank, 1973; Bakker, 1980).

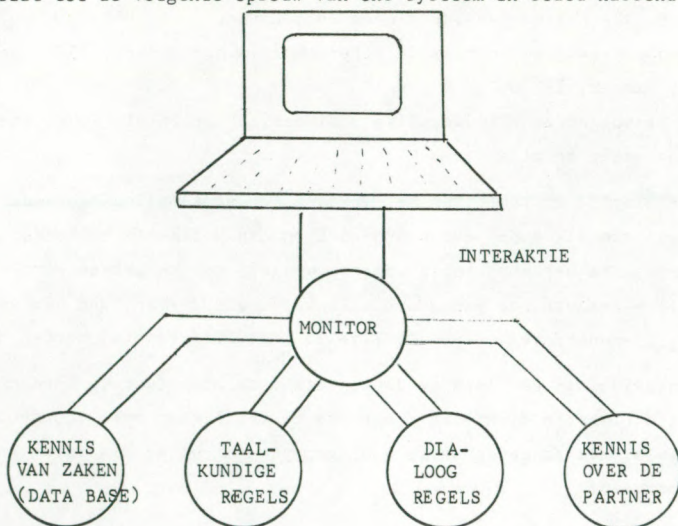
Voor de volgende ontledingsnivo's dienen echter in elk geval voorzieningen aanwezig te zijn:

- morfologie: op basis van het lexicon (de verzameling 'gekende' woorden) dient aan elk woord een woordsoort en een betekenis te worden gehecht. Hiertoe is het soms nodig woordvormregels toe te passen, immers, we willen dat van een werkwoord alle vervoege vormen, van een zelfstandig naamwoord het meervoud, alle samenstellingen etc, worden herkend.
- syntaxis: van een ingevoerde zin dient de structuur te worden vastgesteld. Daartoe dient het programma te beschikken over woordvolgorde-regels, die aangeven welke woordgroepen tot welke zinsdelen en zinnen samengaan.

- semantiek: aan woorden, zinsdelen en zinnen dient een betekenis te worden toegekend. Het lexicon geeft slechts de grondbetekenis(sen) van elk woord apart. De semantische regels dienen, op basis van de woordbetekenissen en de kontekst, de juiste interpretatie aan samenstellingen, woordgroepen, zinnen en grotere gehelen toe te kennen.
- pragmatiek: niet altijd zeggen sprekers precies wat ze bedoelen. Ons programma moet inzien dat achter een vraag ('kunt u mij misschien zeggen of ...') in feite een opdracht ('geef mij ...') verborgen zit.
- makrostructuur: zowel kwa vorm als inhoud worden taaluitingen (mede) bepaald door wat voorafging (denk aan het gebruik van voornaamwoorden) en door algemene en onderwerpgebonden kennis (die niet expliciet wordt gemeld, maar als bekend wordt verondersteld). Bovendien zijn er tussen uitingen allerlei logische (causale, temporele etc.) relaties. Aan elke tekst ligt een structuur ten grondslag die dergelijke verbanden verantwoordt.
- superstructuur: een dialoog is niet een willekeurige verzameling zinnen, maar heeft een zeker verloop (opening met begroeting; bepaling van het onderwerp van gesprek; beurt-afwisseling; op een vraag volgt een antwoord of een wedervraag; etc.).

De genoemde procedures zijn niet lineair of hiërarchisch geordend, maar zijn over en weer van elkaar afhankelijk en staan dan ook in voortdurende verbinding met elkaar via een monitorprocedure.

Dit leidt tot de volgende opbouw van ons systeem in statu nascendi:



afb. 6: toekomstig systeem

4. Slot

De computer is er en kan ons bepaalde werkzaamheden uit handen nemen, waarvan wij, de gebruikers, vinden dat het verantwoord is. Voorlichting, ook die over het gebruik van computerprogrammatuur, geschiedt in het algemeen in de vorm van een zeer komplekse dialoog, gebaseerd op kennis van zaken en de kommunikatieve technieken waarover mensen beschikken.

De kennis over deze beide laatste aspecten is, voor zover dergelijke processen al ooit geheel geformaliseerd zullen kunnen worden, laat staan gealgoritmiseerd, in dit stadium stellig nog te summier om aan iets als volledige simulatie van een voorlichtingssituatie te kunnen denken.

Toch hopen wij aannemelijk te hebben gemaakt, dat een niet-triviaal deel van de voorlichtingstaak, met name het meer routinematige, documentaire aspekt ervan, met enig sukses in een computerprogramma kan worden ondergebracht, en dat er uitzicht is op een wezenlijke uitbreiding daarvan.

Literatuur

- Bakker, D.: Over tekstwetenschap; TC-rapport 80-3; (1980) pp.12-64
- Bakker, D., U. Brouwer en P. Debets: "TCINFO: een automatische programma detector". Methoden en Data Nieuwsbrief 5 (1980) pp. 84-97.
- Brouwer, U. en P. Debets: Evaluatie van statistische programmatuur; i.h.b. m.b.t. MDS. Kwantitatieve Methoden 4 (1981) pp.109-134.
- Hoogstraten, J.; De machteloze onderzoeker. Boom Meppel, 1979.
- Lehnert, W.G.: The process of question-answering; Hillsdale 1978.
- Schank, R.C. en K.M. Colby: Computer models of thought and language; San Francisco, 1973.
- Soudijn, K.: Dilemma's in sociaal-wetenschappelijk onderzoek. Boom Meppel, 1976.
- Zee, H. v.d.: Data analyse. Boom Meppel, 1979.