

DE PROGRAMMATUUR SECTIE

De plannen van de programmatuur groep.

In december 1975 werden binnen de nederlandse universiteiten de eerste contacten gelegd, die uiteindelijk zouden leiden tot de groep die deze methoden en data nieuwsbrief (M.D.N.) redigeert. De initiatiefnemers stond voor ogen een aantal mensen te verzamelen die het braukliggend terrein tussen methodologie en programmatuur voor de sociale wetenschappen konden exploreren. Alhoewel het zowel op het gebied der methodologie als op het gebied der programmatuur beslist niet aan activiteiten ontbreekt, ervaren professionele programma-gebruikers de afstemming van beide gebieden op elkaar als uiterst gebrekkig. Onder professionele programma-gebruikers verstaan we met name diegenen die verantwoording dragen voor aan- of afwezigheid van programmatuur, de kwaliteit ervan, de voorlichting (consultatie, dokumentatie e.d.) daarover, etc. De eerste deelnemers wilden vooral komen tot snelle uitwisseling van informatie, programma-tuur en dokumentatie en, alhoewel de uitgroei naar andere gebieden (modellen en data) de aandacht soms afleidde, deze praktische benadering is een vruchtbare gebleken.

In de eerste mededeling van GENEGE (Genootschap van Nederlandse Gebruikers, de huidige programma-groep) werd gesproken over "belangenbehartiging" met als illustratieve toelichting

"- uitwisseling programmatuur

- aanschaf programmatuur

- dokumentatie

- research t.a.v. - toegankelijkheid van programmatuur

- betrouwbaarheid van programma's

- inhoudelijke wijzigingen van programma's

- schrijven van nieuwe programma's".

In dit stuk wil de programma-groep een aantal punten wat nader specificeren met betrekking tot de "research t.a.v. betrouwbaarheid en inhoudelijke wijzigingen van programma's". Dat een dergelijke opsomming slechts uitblinkt door onvolledigheid, moge daarbij vergoelijkjt worden door het feit, dat gebaande wegen hier vaak ontbreken.

Applikatie-talen

Waar zo'n tien jaar geleden nog voornamelijk met ad-hoc programmatuur werd gewerkt wordt tegenwoordig in toenemende mate met algemene programma-pakketten gewerkt, waarbij de gebruiker zelf verbijzonderend kan optreden.

Bij de veelvoud van mogelijkheden die dergelijke pakketten voor bieden is het noodzakelijk dat de gebruiker een krachtige, relatief eenvoudige besturing ten dienste staat. Te dien einde ontwikkelden de diverse pakket-ontwerpers 'taaltjes' die de gebruiker in staat stellen de controle over het desbetreffende pakket ter hand te nemen. Ofschoon enige uniformiteit, als het gebruik van soms identieke keywords, niet ontkend kan worden, wordt de ontwikkeling van dergelijke 'controle-talen' nog steeds gekenmerkt door een voor gebruikers soms verwarrende diversiteit. Door de samenstelling van de programma-groep mag verwacht worden dat dit gezamenlijke gebruikersbelang onderkend wordt. Daarbij denken we niet in de eerste plaats aan het ontwikkelen van een 'esperanto voor pakketten'. Met instemming citeren we van Hoboken en Niemöller:

"... in a well defined and universal data analysis package there need to be a clear correspondence between the terminology of the methodological theory and the control language of the package. The control language can be used as a computer language to process pre-implemented applications. This aspect of control languages for application packages seems so very important to us, that we want to introduce the concept of application language, an artificial language for the activation of a priori programmed algorithms in terms of the definitions and parameters of an application of methodological theory.

We like to use the term application language instead of e.g. control language or user language to stress the specific role of the type of computer languages we consider: performing applications of scientific research on a computer. The term control language does not make sense, since every computer language is designed to control tasks or operations of automata. The same holds for user or user-oriented languages. ALGOL, BASIC and also a greater part of FORTRAN, are user-oriented quasi-natural languages, but for a totally different group of users and for different tasks: step by step programming of algorithms by programmers. What really does matter is the type of user - the target group - the language is designed for. Application languages should be developed for users in fields of applied scientific research. ..."

"...In the sense of selection of pre-programmed tasks with specification of parameters, application languages are similar to job control languages, a special branch of the computer languages. However, in the hierarchy of programming languages there also seems to be a place for application languages: we think they can be called 'higher macro languages' for the following reasons:

Pure and elementary programming languages are still machine languages or assemblers: every instruction represents an operation of the computer system. But machine languages are written for machines and better tools

are being developed for people, since programmers want to formulate their problem in a sequence of statements. Compilation of a program in a higher programming language, however, produces a program in machine language - every statement is transformed in one or more instructions, which means operations which will be processed.

Macro languages are the result of a different development in programming languages. There are sequences of operations which have to be processed many times in a standard way, only dependent on some parameters to be selected by the programmer. Such sets of instructions - comparable to subroutines - are called macros and actually contain a pre-programmed algorithm, which is processed according to the specified parameters. What the user of an application language actually does is programming his problem by means of language specifications, with which he activates the required blocks of instructions (the procedures of the package); from this point of view the setup, with which the user activates the package according to his problem, is a user program in a high macro language for his applications."

(ontleend aan: Willem H. van Hoboken en Kees Niemöller, Package Development, General Remarks and a Contribution to the Further Development of SPSS, 1976)

Methodologische en programma-technische problemen.

Na dit wat breder uitgemeten eerste punt willen we in kort bestek nog een aantal zaken releveren die o.i. op het terrein van de programma-groep liggen, maar zonder expliciete signalering wellicht te weinig aandacht krijgen.

Een aantal van de hieronder te vermelden punten hebben veel te maken met het simpele bestaan van statistische pakketten. NIE, een der auteurs van SPSS, constateert b.v. dat 'ease of access often means over access' terwijl tevens 'increased statistical capability' niet altijd gepaard gaat met 'increased statistical knowledge'. Alhoewel het hier om een gebruikersprobleem gaat, dat grotendeels door de gebruikers zelf opgelost dient te worden, lijkt het niet onmogelijk vanuit de pakketten zelf een positieve bijdrage te leveren. Andere punten hebben te maken met de relatie tussen programmeur en gebruiker. 'Efficiency' heeft voor een gebruiker vaak een heel andere betekenis dan voor een programmeur. Wij noemen de volgende negen punten:

- wiskundige mogelijkheden en methodologisch begrip houden vrijwel nooit gelijke tred. We bedoelen daarmee te zeggen, dat wat wiskundig mogelijk is, lang niet altijd methodologisch zinvol is. In de

gevallen waarin programma's resultaten opleveren die mathematisch correct zijn maar niet interpreteerbaar zijn in termen van het onderzoekssubject, bestaat het gevaar dat mathematische kwantiteiten een geheel eigen leven gaan leiden.

- methodologische foutmeldingen: Vele programma's gaan uit van een of meer assumpties met betrekking tot de kwaliteit van de data. Van de gebruikers wordt verwacht dat ze toetsen of hun data voldoen aan vereisten als homoscedasticiteit, normale verdeling, lineariteit etc. Ofschoon dergelijke zaken als regel niet al te moeilijk zijn, is het voor de doorsnee gebruiker meestal een bewerkelijke omweg, die danook zelden bewandeld wordt. Naar onze mening zouden dergelijke toetsen zoveel mogelijk beschikbaar moeten zijn waar dat nodig is, d.w.z. in de programma's zelf. Het blijft daarbij de vraag of een programma door moet rekenen wanneer aan belangrijke assumpties die aan het gebruikte model ten grondslag liggen niet of, nog moeilijker, niet geheel voldaan is. In ieder geval dient in zo'n geval een duidelijke waarschuwing in de output te verschijnen, zodat de gebruiker voor twee telt. Dit brengt ons tot het meer algemene punt van de methodologische (fout)meldingen. Toegenomen statistische mogelijkheden zonder gelijktijdig toegenomen statistische kennis vormt een van de meest gesignaleerde gevaren uit de dagelijkse praktijk van consultatie-afdelingen. We zijn er dan ook van overtuigd dat meer, véél meer, methodologisch gerichte meldingen noodzakelijk zijn. Het leereffect dat van zo instructief mogelijk gestelde meldingen uit gaat, kan ons helpen negatieve aspecten van programmatuur te verminderen.
- little jiffy is een term door Chester Harris gegeven aan het voor beginners ijzersterke factor-analyse recept: 'principal components with associated eigenvalues greater than one followed by normal varimax rotation'. In vele programma's zijn, d.m.v. het default-mechanisme, vele vergelijkbare Little-jiffies ingeslopen. Vele beginners kunnen geen doordachte keus maken uit de mogelijkheden die een programma biedt. Het default mechanisme zorgt dan voor een keuze die niet noodzakelijkerwijs dezelfde doordachte methodologische basis heeft als de oorspronkelijke little jiffy.
- missing data vormen in de sociale wetenschappen een ernstig probleem. Statistische programmatuur biedt in toenemende mate opties voor missing data. De drie meest gebruikte, 'inclusion-, listwise deletion- en pairwise deletion- van missing data', vormen vaak geen ideale oplossing. Ook een vierde wel gebruikte optie, substitutie van het gemiddelde, stuit in vele gevallen op bezwaren. Het missing data probleem wordt

te weinig onderkend.

- toeval en het GIGO-syndroom: Nie's eerder geciteerde opmerking dat 'ease of access often means over-access' wordt door hemzelf nog wat verduidelijkt door wat hij noemt 'grand fishing expeditions, substituting the crudest form of empiricism for the careful interactions of concepts, hypotheses and data-analysis'. Dergelijke expedities worden door statistische pakketten en de computeromgeving vergenakkelijkt omdat ze georganiseerd zijn '... to make it economically attractive to produce as much output as possible on a single pass of the data'.
Dit voert vaak tot het zgn. gigo-syndroom, welk acronym staat voor GARBAGE IN - GARBAGE OUT. Terecht wijst Nie er vervolgens op dat bij de creatie van dergelijke massieve hoeveelheden output (b.v. kruis-tabellen: alles tegen alles) het toeval voor significante resultaten zorg kan dragen.
- foutmeldingen: Bij de toenemende complexiteit der statistische programmatuur is een uitvoerige, heldere foutmelding noodzakelijk. Vele pakketten en programma's hebben een min of meer uitvoerige foutmelding, die echter al te vaak zeer gecomprimeerd is of in onbegrijpelijk jargon blijft steken. Geconstateerd moet daarbij worden, dat het merendeel der meldingen betrekking heeft op de syntactische scan van de applicatie-taal. Daarbij wordt de behoefte aan meer methodologisch gerichte meldingen niet bevredigd.
- output: In het algemeen is de output minder toegesneden op de wensen van de gebruiker en weerspiegelt voornamelijk de wens van de programmeur alle relevante rekenresultaten netjes uit te serveren. Overleg tussen programmeur en ervaren gebruiker leidt vaak tot verrassende wijzigingen in de lay-out, waardoor het resultaat beter in het analyseproces van de gebruiker past.
- documentatie: Ter versterking van het positieve effect dat uitgaat van een goed gedocumenteerde output, dient de gebruiker de beschikking te hebben over hoogwaardige documentatie.
Een aantal leden van de programma-groep vindt dit zo'n belangrijke zaak, dat hier apart op zal worden ingegaan, terwijl ook een aantal initiatieven op dit terrein ontwikkeld wordt. Op dit moment verdagen we dit punt naar een volgend nummer.
- nieuwe ontwikkelingen: Tot slot willen we nog wijzen op de soms onvermijdelijke kloof tussen statistische pakketten en nieuwe ontwikkelingen. Ontwikkelingen van vooral de grote, veel gebruikte statistische pakketten

vormt een dermate grote onderneming dat het resultaat meestal nogal behoudend uitvalt: slechts algemeen bekende en geaccepteerde methodes zullen worden opgenomen. Dit kan er licht toe leiden dat programma's met meer geavanceerde methoden, die niet zo makkelijk toegankelijk zijn als de standaard pakketten, niet die aandacht krijgen die noodzakelijk is voor een gezonde ontwikkeling van sociaal wetenschappelijk research. Het ontstaan van gespecialiseerde pakketten, zoals het cluster-analyse pakket CLUSTAN, kan dit gevaar zeker verminderen, alhoewel hetzelfde probleem op een hoger niveau weer kan optreden.

De groep die zich thans bezig houdt met de raakvlakken van methodologie en programmatuur, heeft als algemene doelstelling het stimuleren van samenwerking en uitwisseling op het gebied van documentatie, methodologische verbetering en beschikbaarheid van programmatuur voor sociale wetenschappers.

De hierboven genoemde punten als applicatietalen, methodologische aspecten, toegankelijkheid in gebruik, maar ook een punt als auteursrechten van programmatuur zijn bedoeld als specificaties hiervan.

In de periode die voor ons ligt willen we proberen een aantal programma's of programmapakketten op deze genoemde punten te beoordelen en hierover consumentengids-achtige rapporten uit te brengen.

Voorts willen we de lezer informeren over nieuw verschenen programmatuur en bijbehorende documentatie.