

Systematiek en uniformiteit in de mens/machine-communicatie

Citation for published version (APA):

Beukering, van, L. H. T. M., & Banens, J. P. A. (1985). *Systematiek en uniformiteit in de mens/machine-communicatie: ervaringen bij de inzet van een standaard gebruikersinterface*. (DCT rapporten; Vol. 1985.032). Technische Hogeschool Eindhoven.

Document status and date:

Gepubliceerd: 01/01/1985

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Systematiek en uniformiteit in de mens/machine-communicatie.

Ervaringen bij de inzet van een standaard
gebruikersinterface.

inhoud:

0. Samenvatting
1. Inleiding
2. Mens/machine-communicatie
3. Eisenpakket
4. Implementatie
5. Ervaringen.
6. Verdere ontwikkelingen.
7. Literatuur.

ing. L.H.Th.M. van Beukering,
Ir J.P.A. Banens
Technische Hogeschool Eindhoven,
Afdeling der Werktuigbouwkunde.
Januari 1985.



O. Samenvatting.

Standaardisering van de Mens/Machine-communicatie door het gebruik van een Kommando-interpretator.

De afdeling der Werktuigbouwkunde van de Technische Hogeschool Eindhoven heeft een meerjarige ervaring op het gebied van de ontwikkeling van interaktieve software. In de achter ons liggende jaren is dientengevolge een filosofie met betrekking tot de Mens/Machine-communicatie ontstaan, die de aanleiding was tot de ontwikkeling van het software-pakket COIN.

COIN, (COmmand - INterpreter) is een verzameling routines die de kommando-afhandeling van een programma volledig voor zijn rekening neemt. Het toepassen van COIN verbetert de kwaliteit van software-produkten wezenlijk, onder andere doordat door het gebruik van deze routines een systematische opzet van een programma bewerkstelligd wordt. Een programma dat met COIN is uitgerust heeft de eigenschap dat een beginnend gebruiker volledig wordt begeleid en dat een ervaren gebruiker geen last heeft van die begeleiding.

COIN kent een hulpfunctie, die qua uitwendige kenmerken, omgevingsafhankelijk is; op ieder niveau wordt voor dat niveau relevante informatie aangeboden. De commandostructuur en de hulpfunctie zijn geen deel van de applicatie zelf. Hierdoor kunnen wijzigingen aan deze structuur doorgevoerd worden zonder dat het programma zelf gewijzigd behoeft te worden.

COIN geeft alle mogelijkheden om applicaties van een betrouwbaar en snel dialoogsysteem te voorzien. Kenmerkend voor het gebruik van een dergelijke interpretator is het feit dat de programmeerinspanning wezenlijk wordt verminderd.

In deze voordracht wordt het concept toegelicht en worden de ervaringen van de afdeling der werktuigbouwkunde met betrekking tot het gebruik van dit pakket doorgegeven.

0. Abstract.

Standardization of the man/machine communication by the application of a command-interpreter.

At the Eindhoven University of Technology several years experience exists in developing interactive software. Concerning the man/machine communication a philosophy has evolved, leading to the development of COIN. COIN (COmmand - INterpreter) is a set of routines, which completely takes care of the command processing of a program. Employing this facility essentially improves the quality of the developed software. One of the reasons for this is the fact that use of the routines results in a systematic program organization. One of the COIN characteristics is that it offers full support to the novice, while being a useful expedient for the professional user. The support function can be employed in every environment and the support is a function of the current environment itself.

The command processing and the support function are not incorporated in the application program, which makes it possible to change this function without maintenance of the program. COIN offers the possibility to provide demanding, individual application programs with an effective dialog system, at the same time considerably reducing the necessary programming effort.

In this lecture the concept of COIN is explained and user-experiences at the university are discussed.

O. Résumé

Standardisation de la communication homme-machine par l'utilisation de logiciels du type "Command Interpreter".

L'Université de Technologie d'Eindhoven a, depuis plusieurs années, accumulé une expérience dans le développement du software interactif.

Une philosophie qui concerne la communication homme-machine s'y est développée, qui a donné naissance au logiciel "COIN". COIN (COmmand INterpreter) est un ensemble de logiciels qui prend totalement en charge le déroulement d'un processus de programmation. L'application de COIN améliore en profondeur la qualité des logiciels par l'organisation systématique du programme d'application.

Une des caractéristiques de COIN est d'être un soutien complet pour le débutant, tout en n'étant d'aucune gêne pour le professionnel. La fonction de support de COIN dépend de l'environnement : il fournit à chaque niveau les informations particulières nécessaires.

Le logiciel COIN tout en étant logiciel-support et d'aide à la programmation ne modifie aucunement le programme de l'application, ce qui permet des adaptations des interventions sans en modifier le programme principal.

COIN offre toute possibilité de doter les programmes d'un système de dialogue rapide et fiable. L'utilisation d'un tel "interpreter" diminue d'une façon considérable les difficultés de programmation.

Cet exposé explique le concept de COIN et fait part des expériences qu'a fait l'Université d'Eindhoven de ce système.

0. Kurzbeschreibung.

Standardisierung der Mensch/Maschine-Kommunikation durch Anwendung einer Kommando-Interpreter.

Die Technische Universität Eindhoven in den Niederlanden hat eine mehrjährige Erfahrung in dem Bereich der Entwicklung von interaktiver Software. Während dieser Jahre ist eine Philosophie entstanden über die Mensch-Maschinen-Kommunikation, welche der Anlass war zu der Entwicklung von COIN.

COIN, (COmmand-INterpreter) ist eine Sammlung von Routinen die den Kommando-Ablauf eines Programmes vollständig übernimmt. Die Benutzung dieses Werkzeuges verbessert die Qualität der Software-Produkten wesentlich, u.a. weil die Routinen einen systematischen Aufbau des Programmes bewirken. Die Eigenschaften von COIN sind so, dass der Anfänger vollständig begleitet wird und auch der 'Profi' voll auf seine Kosten kommt. Die Hilfefunktion ist umgebungsabhängig und die Reihe von möglichen Kommandos ist auf jedem Niveau verfügbar. Das Kommandoverfahren und die Hilfefunktion sind ausserhalb der Applikation gehalten, wodurch Änderungen dieser Funktion ohne Pflege des Hauptprogrammes durchgeführt werden können.

COIN bietet alle Möglichkeiten, um anspruchsvolle, individuelle Anwendungen mit einem sicheren und schnellen Dialogsystem zu versehen und dabei den sonst benötigten Programieraufwand wesentlich zu reduzieren.

In diesem Referat wird das Concept erklärt und werden die Erfahrungen des Einsatzes an der Universität weitergegeben.

1. Inleiding.

In 1979 is binnen de afdeling der werktuigbouwkunde van de Technische Hogeschool Eindhoven een lijn geformuleerd waarin ter ondersteuning van onderwijs en onderzoek aan "de computer" een specifieke rol toegekend is. Direct daarna zijn investeringen gedaan die het mogelijk maakten om door middel van een homogeen gebruik van mainframe, supermini en microcomputer dit beleid gestalte te geven. Het huidige beleid en de nu heersende filosofie zijn nog steeds een rechtstreekse afgeleide van de destijds geformuleerde uitgangspunten. Voor de gebruiker van rekenfaciliteiten springt in dit beleid het streven naar een universele gebruikersomgeving sterk naar voren. Een van de aspecten hiervan is de definitie van de communicatie tussen de gebruiker en een computersysteem.

In deze lezing wordt ingegaan op een aantal aspecten van dit ontwerp en worden de ervaringen van zowel de gebruikers alsook van de software-engineers doorgegeven.

2. De mens/machine-communicatie.

Bij gedachten over de totaalproblematiek van de mens/machine-communicatie hebben we steeds te maken met twee aspecten. Op de eerste plaats heeft deze problematiek betrekking op de gebruiker

van informatiesystemen; op degene die in communicatie treedt met een systeem. Daarnaast komen er aspecten aan de orde die van belang zijn voor de ontwerpers van dergelijke systemen. Bij het ontwerp van een communicatiemogelijkheid zullen beide aspecten aandacht moeten krijgen.

2.1 De gebruiker.

Algemeen wordt gesproken over de mens/machine-communicatie, hetgeen reeds aangeeft dat de eigenschappen van het individu dat de communicatie voert, voorop dienen te staan. In de literatuur zijn vele modellen ter beschrijving van de communicatie te vinden. Een goede basis voor het bepalen van de gedachten over deze communicatie vormt het IFIP-model [1],[2]. Hierin wordt uitgegaan van een algemene communicatie tussen twee partners.

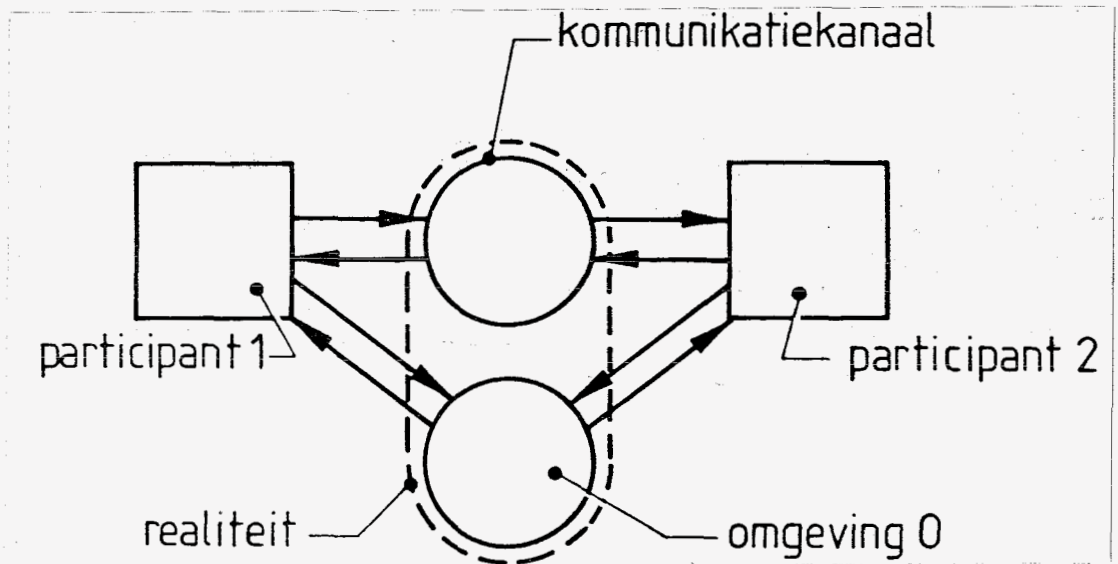


fig. 1. Communicatie in basisvorm.

Tevens wordt gesteld dat communicatie steeds plaatsvindt in een **communicatieomgeving**, die een deel is van een totale omgeving, **realiteit** genoemd. De communicatie vindt plaats over een **communicatiekanaal**, waarbij steeds een interactie aanwezig is met de communicatieomgeving. Kijken we naar de structuur van de

participanten in een communicatie dan komen daarbij de in figuur 2 aangegeven aspecten aan de orde:

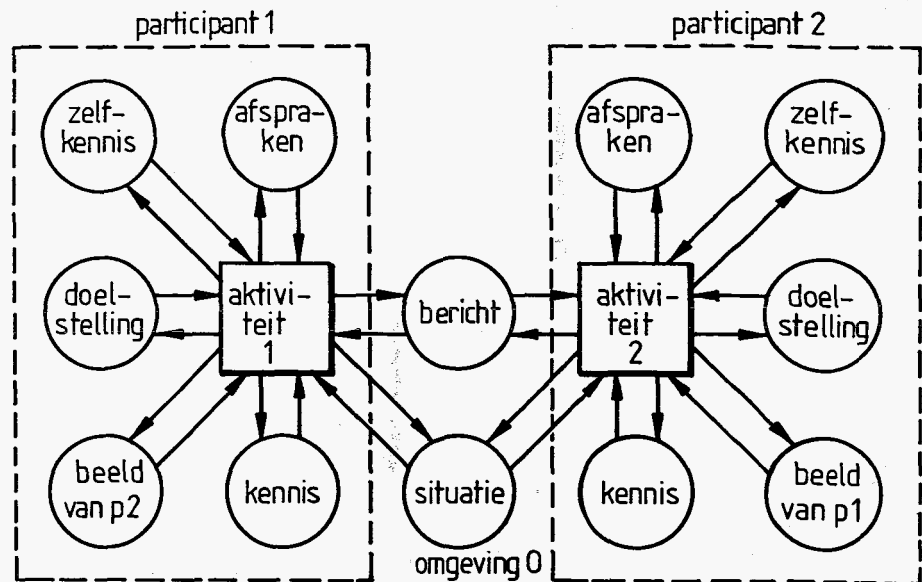


fig. 2. De structuur van de participanten.

Als we communicatie definiëren als een gecoördineerd symbolisch handelen tussen twee of meerdere participanten met gebruikmaking van een medium dan komen we tot de volgende opmerkingen:

1. De vorm van de communicatie is afhankelijk van de kennis van de participanten.
2. De inhoud van de communicatie is afhankelijk van het doel dat de participanten wensen te bereiken.
3. De begeleiding van het communicatiesysteem dient afgestemd te zijn op het beeld dat de partners van elkaar hebben.
4. Het bericht dat verzonden wordt zal een relatie moeten hebben met de vermeende zelfkennis van de ontvangende participant en in relatie moeten staan met de communicatieomgeving.
5. Communicatie op een bepaald tijdstip heeft een relatie met hetgeen in het verleden reeds plaatsgevonden heeft.

Om een dergelijk model en de daarbij behorende kenmerken te kunnen vertalen naar een communicatie tussen een mens en een machine kunnen we gebruik maken van het IFIP-model van een dialoogomgeving.

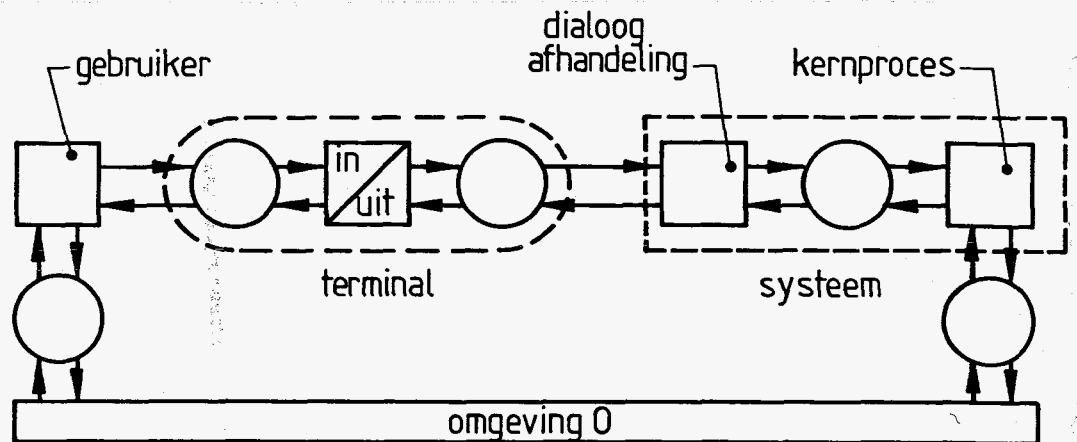


fig. 3. Het IFIP-Model van een dialoogomgeving.

Dit model ontstaat uit het in figuur 2 gegeven structuurdiagram door specificatie van het communicatiekanaal en door aan de partner P1 de eigenschappen toe te kennen van de gebruiker die een dialoog voert met een computersysteem. Om in hoofdstuk 2.2. nader te verklaren redenen is de partner P2, het computersysteem, onderverdeeld in twee kenmerkende subsystemen. Hierdoor is het construeren van een dialoogsysteem geworden tot de constructie van een "partner P2" die maximaal voldoet aan de eisen die door de participant P1 opgelegd worden. Deze eisen zijn direkt te destilleren uit de eigenschappen van P1 en de dialoogomgeving zoals die in figuur 2 aangegeven zijn.

Eisen die gesteld worden aan een dialoogsysteem zijn bijvoorbeeld:

1. Het communicatiesysteem dient een help-functie te bezitten die op verzoek van de gebruiker geactiveerd wordt.
2. De helpfunctie behoort informatie te verschaffen die afhankelijk is van de momentane communicatieomgeving.
3. Het communicatiesysteem dient suggestie's te doen aangaande het antwoord op de eventueel te stellen vragen.
4. Taal, formattering en structurering dienen afgestemd te zijn op de eigenschappen van de gebruiker. Daar waar dit door de applicatie niet geëist wordt dient geen enkele eis gesteld te worden aan de vorm waarin een communicatie-item aangeboden wordt.
5. Een opdracht aan het communicatiesysteem dient door het communicatiesysteem gezien te worden als een element uit een volledig pad in een structuur. In deze context wordt onder structuur een set van vragen en commando's, tussen welke een één-eënduidige relatie bestaat, verstaan.
6. Het dient de gebruiker toegestaan te zijn om, op elke wijze die niet in strijd is met de door de structuur opgelegde eisen, zich binnen de structuur te verplaatsen.
7. Communicatiefouten, zowel syntactische als semantische dienen adequaat afgehandeld te worden, hetgeen betekent dat een omgevingsafhankelijke foutmelding gegeven dient te worden en dat daarnaast de status van de communicatie gelijk dient te zijn aan de voor het optreden van de fout relevante status.
8. De gebruiker dient in de gelegenheid te zijn om, zonder tussenkomst van het dialoogsysteem, die communicatie te voeren die hij wenselijk acht. Dit houdt ondermeer in dat sequenties van commando's en antwoorden, als één opdracht ingegeven, toegestaan dienen te worden.
9. De gebruiker dient in de communicatie steeds de initiatiefnemer te zijn.

Het onderzoek dat verricht wordt op het gebied van het gebruik van natuurlijke talen bij de instructie van computersystemen is voornamelijk gericht op het omzetten van expressie's in natuurlijke talen naar constructie's in formele talen.

Bij de dialoog heeft men daarentegen te maken met een ander aspect; hier speelt slechts de herkenning van elementen van een taal een rol. Bij een gegeven proces geeft een bepaald taalelement aan dat een specifiek algoritme uitgevoerd dient te worden. De taalelementen zelf dienen dicht bij de dialoogvoerder te staan, hetgeen betekent dat ze moeten stammen uit het vakgebied waarop de applicatie betrekking heeft en kenmerken moeten hebben van een natuurlijke taal die de gebruiker goed verstaat.

2.2. Het ontwerp van een dialoogstelsel.

Een rekenautomaat volgens het von-Neumann principe is een werktuig dat in eerste instantie geconstrueerd is om, gestuurd door een programma, de inhoud van een geheugen te veranderen. Immers de geheugeninhoud is representatief voor de variabelen die we volgens een bepaald algoritme van waarde willen laten veranderen. We zullen het proces dat hiervoor verantwoordelijk is het **kernproces** noemen. Deze gedachtengang impliceert dat de communicatie tussen gebruiker en kernproces gescheiden afgehandeld wordt. We noemen het proces waarin de afhandeling van de communicatie met de gebruiker plaats vindt het **dialoogproces**. De hier gegeven strikte scheiding tussen deze processen betekent niet dat de dialoogafhandeling door een andere processor uitgevoerd moet worden; echter wel dat het andere algoritmen zijn die voor de dialoogbegeleiding zorgen.

Deze scheiding heeft de volgende voordelen:

- We behoeven bij het ontwerp van softwaresystemen geen beslissingen te nemen met betrekking tot de plaats waar de dialoogafhandeling plaats vindt. We hebben de vrije keuze tussen elementaire terminal en geavanceerd workstation.
- Het communicatiesysteem heeft veelal een veel groter toepassingsgebied dan de applicatie zelf. Door scheiding kan deze eigenschap veel eenvoudiger benut worden.

Bij het ontwerp van een dialoogsysteem kan niet uitgegaan worden van wathematische communicatietheorieën omdat hierbij steeds verondersteld wordt dat het gedrag van de participanten berekenbaar is en zaken zoals doelstelling en omgeving vaste gegevens zijn. Formele communicatie, zoals men die vindt bij datacommunicatieprotocollen, voldoen wel aan deze eisen en zijn daardoor op basis van voornoemde theorieën algoritmiseerbaar. De pragmatiek die leidt tot goede resultaten bij de definitie van een vorm van mens/machine-communicatie is sterk verwant aan een aanpak die in de organisatiepsychologie gehanteerd wordt. Bij deze aanpak wordt getracht om uit een bepaalde communicatie de cognitieve processen te onderscheiden met als doel deze vervolgens van een adekwate afhandelingsprocedure te voorzien [3].

3. Eisenpakket.

Bij het samenstellen van het eisenpakket speelt de dialoogomgeving een belangrijke rol. De opsomming, zoals die hier gegeven wordt, is afgeleid van de situatie op de afdeling der Werktuigbouwkunde van de T.H.E.

De eisen die aan een dialoogproces gesteld worden vanuit het oogpunt van de gebruiker worden beschreven door de punten die aangehaald zijn bij de beschrijving van de communicatie zelf in hoofdstuk 2.1. Voor de applicatieprogrammeur zijn de volgende aspecten van belang:

- Zoals in de inleiding reeds aangegeven is wordt binnen onze afdeling gebruik gemaakt van een grote verscheidenheid aan computersystemen. Daar de gebruikersomgeving niet afhankelijk mag zijn van het gebruikte systeem, dient gekozen te worden voor een opzet die op alle systemen te implementeren is. Gekozen is daarom in eerste instantie voor een implementatie geschreven in standaard FORTRAN IV.
- De programmeur moet in de gelegenheid zijn om eisen te stellen aan het resultaat van de dialoog. Het type van het resultaat en de grootte van de ingevoerde waarde dient in die gevallen waarin dit relevant is, een parameter van de gestelde vraag te zijn. Het dialoogsysteem dient onder alle omstandigheden een resultaat af te leveren dat in overeenstemming is met de eisen van de programmeur.
- De applicatieprogrammeur mag niet in aanraking behoeven te komen met het specifieke van de programmeertaal waarin het dialoogprogramma geschreven is. Met name input/output en formatering zijn hierin belangrijke elementen.
- De verzorging van de communicatie en de teksten die daarbij een rol spelen dient buiten het kernproces plaats te vinden.
- De applicatieprogrammeur dient de commandostructuur vast te kunnen leggen.

4. De implementatie.

De subroutinebibliotheek die ten behoeve van het verwezenlijken van de in de voorgaande aangegeven functie is geconstrueerd heeft de naam COIN (COMmando INTERpretator) gekregen. Het concept hiervan is ontworpen door ir J.P.A. Banens [4].

De implementatie die hier besproken wordt moet gezien worden als de basisimplementatie, die op alle systemen aanwezig is. Gezien de eigenschappen van de in de afdeling aanwezige randapparatuur (eenvoudige terminals) is gekozen voor een opzet, waarbij de afhandeling van het dialoogproces ook plaats vindt op het "centrale" computersysteem. In dit hoofdstuk wordt geen uitputtende uiteenzetting gegeven van alle "in en outs" van COIN. Er wordt aangegeven hoe COIN kan reageren naar een gebruiker en hoe een programmeur er mee om kan gaan.

4.1. De gebruiker

De gebruiker zit achter een terminal en wil met behulp van een applicatieprogramma een nader te omschrijven doel bereiken. Als voorbeeld nemen we een programma dat op een personenbestandje opereert. Veronderstel dat onze gebruiker wil weten hoeveel personen er in dat bestandje zitten die in 1981 of later geboren zijn. We gaan er vanuit dat hij niet weet hoe hij dat moet doen, maar dat het applicatieprogramma die vraag wel kan beantwoorden. Nadat de gebruiker de applicatie gestart heeft drukt hij op de RETURN-toets en krijgt dan bijvoorbeeld te zien:

?
H - HELP
SE - SELECTEER
SO - SORTEER
A - AANTAL
L - LIST
Q - QUIT

OK>

Als basisprompt is in dit voorbeeld gekozen voor "OK>". Achter deze prompt mag de gebruiker weer iets tikken. Dat mag een van de beschikbare commando's (HELP, SELECTEER etc.) zijn of een vraagteken. In dit laatste geval vraagt hij om meer begeleiding. Kiest hij voor SELECTEER (af te korten tot minimaal SE) dan verschijnt er:

?
N - NAMEN
G - GEBOORTEDATA
S - SALARIS

SELECTEER>

Als prompt verschijnt nu de naam van het geselecteerde commando. Stel dat de keuze nu valt op de G van GEBOORTEDATA . Na G en een druk op RETURN volgt dan:

?
V - VANAF
T - TOT_EN_MET

GEBOORTEDATA>

Na de keuze VANAF gemaakt te hebben krijgt hij nu een vraag:

DATUM (811012) :

Mogelijke vragen van de gebruiker:

- Hoe moet je nu een datum verstrekken?
- Wat betekent dat vreemde getal?

Tikt hij een vraagteken in dan verschijnt op het scherm:

U moet een datum intikken in de vorm van een zes-cijferig getal.

De eerste twee cijfers vormen het jaartal in deze eeuw, de volgende twee de maand (januari is 01, februari 02 etc.) en de laatste twee de dag van de maand.

Voorbeeld: 770512 betekent 12 mei 1977

DATUM (811012) :

In het voorbeeld is te zien dat, nadat de uitleg gegeven is, de vraag herhaald wordt. Tussen haakjes staat een suggestie voor het antwoord; wilt U die suggestie overnemen dan is een druk op RETURN voldoende. Onze gebruiker tikt als datum echter 810101 in. Het programma meldt zich nu weer met:

OK>

en de hele gang van zaken kan herhaald worden. Nu kiest onze gebruiker vervolgens voor AANTAL dan verschijnt op zijn scherm:

88 personen geselecteerd

OK>

Indien onze gebruiker de selectie op geboortedatum en aantal wil uitvoeren en het pad reeds kent, dan kan hij deze opdracht ook ingeven door in te tikken:

SE GEB VANAF 810101 A

Het antwoordt "88" volgt in dit geval direkt.

Onze gebruiker kan ook iedere tussenvorm tussen deze twee uitersten gebruiken. Daarnaast staan de wat geroutineerdere gebruiker nog enkele speciale symbolen ter beschikking voor het corrigeren van een gekozen pad door de commando's en voor het onbeantwoord laten van vragen.

4.2 De programmeur.

In deze paragraaf wordt ingegaan op de taken en werkzaamheden van de programmeur. We hanteren hierbij het voorbeeld uit hoofdstuk 4.1. Op de eerste plaats dient de terminologie nader verklaard te worden. COIN onderscheidt commando's en vragen. In paragraaf 4.1 is daar, zonder nadere verklaring, al gebruik van gemaakt.

Een commando is een door COIN herkend woord, bestaande uit letters en/of de onderstreepte spatie. U mag het ook een selectie uit een menu noemen; voor de selectie wordt echter het woord zelf gebruikt zoals dat ook bij commandotalen gebruikelijk is. Voorbeelden uit hoofdstuk 4.1 zijn SELECTEER en AANTAL. Een commando kan subcommando's hebben zoals SELECTEER uit ons voorbeeld. Met een volledig commando bedoelen we een commando met een volledige lijst subcommando's. De structuur, waarvoor gekozen is, is strikt hiërarchisch.

Een vraag is, vanuit programmaoogpunt, een mogelijkheid om van de gebruiker een waarde te krijgen. Dat kan een Boolean (ja/nee), een integer, een real of een tekst zijn. COIN suggereert altijd een antwoord en biedt de gebruiker de mogelijkheid om die suggestie over te nemen.

In het algemeen zal een interactief programma commando's (opdrachten) van de gebruiker accepteren en die vervolgens uitvoeren. Bij de uitvoering van zo'n opdracht kunnen nadere gegevens nodig zijn. In ons voorbeeld heeft het commando SELECTEER GEBORTE DATA VANAF een datum nodig; het commando AANTAL heeft geen nadere gegevens nodig.

Om van COIN gebruik te maken moet de applicatieprogrammeur naast het schrijven van zijn programma een invoerfile voor COIN maken. Kortheidshalve noemen we die file INFCO. In INFCO komen de commando's en de vragen te staan. Daarnaast kunnen hierin "help"-teksten, de teksten die verstrekt worden wanneer de gebruiker een vraagteken intikt, opgenomen worden. Door de structuur van INFCO wordt ook de gewenste hiërarchie vastgelegd.

Het samenstellen van zo'n INFCO file is niet moeilijk en er is geen specialistische programmeerkennis voor nodig. Zorgvuldigheid bij de keuze van commando's en commandostructuur is vanuit gebruiksoogpunt uiteraard wel gewenst. De inhoud van de INFCO file wordt in hoofdstuk 4.2.2 gegeven.

Ieder commando is voorzien van een key, een integer waarde; iedere vraag heeft een (positief) vraagnummer. Een applicatieprogramma maakt uitsluitend gebruik van deze keys en nummers, de teksten van commando's en vragen zijn volledig onzichtbaar. Ook alle hulpverstrekking en behandeling van onbruikbare invoer gebeurt volledig achter de schermen. In hoofdstuk 4.2.1. wordt aangegeven hoe een programmeur het communicatiesysteem kan gebruiken.

4.2.1. Het programma.

In onderstaand deel van een applicatieprogramma gaan we ervan uit dat het commando 'SELECTEER GEBOORTEDATA VANAF' de keywaarde 11 en het commando 'AANTAL' de keywaarde 4 heeft. De vraag 'DATUM' heeft vraagnummer 7. Een stukje uit ons (FORTRAN)-programma, dat volledig is voor wat het voorbeeld uit hoofdstuk 4.1 betreft, luidt dan:

```
C INITIALIZE
      .....
C OPEN INFCO-FILE ON LOGICAL FILE NUMBER 8
(1)      . . . . .
(2)      CALL INCO (8)
C MAIN PROGRAM LOOP
(3)      1 CALL KEYCO (KEY,0)
          GOTO (10,20,30,40,50,60,70,80,90,100,110),KEY
          .....
          .....
C KEY=4 : AANTAL
          40 WRITE (IOUTF,1010) ICOUNT
          1010 FORMAT (I5,22H personen geselecteerd)
          GOTO 1
          .....
C KEY=11 : SELECTEER GEBOORTEDATA VANAF
(4)      110 CALL IQCO (7,IDATE)
          IF (NODATE (IDATE)) GOTO 110
          CALL SELECT (IDATE, ..... )
          GOTO 1
          .....
etc.
```

Opmerkingen:

- (1) : Hier wordt de file INFCO geopend. Deze file moet open blijven gedurende alle services van COIN m.b.t. de hulpverstrekking.
- (2) : INCO initialiseert COIN.
- (3) : KEYCO levert de keywaarde van een volledig commando af.
- (4) : IQCO stelt een vraag die een integer antwoord verlangt. Iedere integer waarde wordt als antwoord geaccepteerd. Daarom volgt in dit programma op de aanroep van IQCO de aanroep naar een, niet tot COIN behorende, logical function NODATE om te verifiëren of het wel een correcte datum betreft.
Ook subroutine SELECT behoort niet tot COIN.

In de gebruikershandleiding van COIN [4] wordt een complete beschrijving van de routines gegeven.

4.2.2. De invoerfile INFCO.

De invoerfile INFCO bestaat uit vier secties: de controlsectie, de commandosectie, de vragensectie en de helpsectie. De vier secties moeten in deze volgorde aanwezig zijn in INFCO. We zullen ze hier achtereenvolgens bespreken. Van alle regels uit INFCO worden slechts de eerste 72 karakters gelezen.

De control-sectie.

De control sectie beslaat de eerste twee regels van INFCO. Hierin wordt controle-informatie gegeven die bij het initialiseren van COIN gebruikt wordt.

De commando-sectie.

De commando sectie bevat alle informatie over de commando's en de commandostructuur. Ieder commando moet op een eigen regel staan. Deze regels worden gelezen met de statements:

```
      READ (INFCO,100) IPHL,KEY,MINCH,(ITEXT(I),I=1,59)
100 FORMAT (3I4, 1X, 59A1)
```

waarin:

IPHL :de "help"-tekst link van het commando

KEY :de key waarde van het commando

MINCH :het minimale aantal karakters voor herkenning

ITEXT :spaties, gevolgd door de commandonaam. De commandonaam mag uitsluitend de letters A-Z en de onderstreepte spatie (_) bevatten.

De commandostructuur, de hiërarchie, wordt vastgelegd door het aantal spaties voor de commandonamen. Een commando voorafgegaan door "n" spaties is een subcommando van het daarvoor gegeven commando met "n-1" spaties. Voor het voorbeeld uit het voorgaande heeft deze sectie de volgende inhoud:

```
      45  1  0  OK
      0  1  1  HELP
      80  2  2  SELECTEER
     234 13  1  NAMEN
     100 10  1  GEBOORTEDATUM
      0 11  1  VANAF
      0 12  1  TOT_EN_MET
      0 14  1  SALARIS
     96  3  2  SORTEER
     60 21  1  NAMEN
      0 22  1  GEBOORTEDATUM
      0 23  1  SALARIS
     220  4  1  AANTAL
      0  5  1  LIST
      0  9  1  QUIT
```

OK is de hoofdnaam, HELP, SELECTEER, SORTEER, AANTAL, LIST en QUIT zijn subcommando's van OK; voor OK, SELECTEER, SORTEER en AANTAL zijn "help"-teksten aanwezig; SELECTEER en SORTEER hebben weer subcommando's; het derde subcommando van SORTEER is SALARIS enz. De minimale afkorting van alle namen is hier een karakter, behalve van SELECTEER en SORTEER; hun minimale afkortingen zijn respectievelijk SE en SO. Er mogen geen subcommandonivo's overgeslagen worden, d.w.z. meer dan een spatie naar rechts inspringen is verboden. In dit voorbeeld wordt "OK" ook als commando beschouwd. De gebruiker behoeft normaliter dit commando nooit te gebruiken; er wordt in het programma van hoofdstuk 4.2.1 steeds om subcommando's van OK (key=0) gevraagd. OK krijgt dientengevolge de functie van prompt op het hoofdniveau. Er mogen meerdere hoofdnamen (namen met slechts een spatie ervoor) toegepast worden. Namen en keys hoeven niet uniek te zijn. De "links" naar de "help"-teksten behoeven niet door de programmeur ingevuld te worden. Daar is een aparte utility "LINKCO" voor.

De vraag-sectie.

De vraag-sectie bevat alle informatie over de vragen. Iedere vraag moet op een eigen regel getikt worden en wordt gelezen met de statements:

```
      READ (INFCO,100) IPHL,IQ,ITYPE,(ITEXT(I),I=1,63)
100 FORMAT (2I4, A1, 63A1)
```

waarin:

IPHL :de "help"-tekst link van de vraag

IQ :het nummer van de vraag

ITYPE :het type van de vraag

B of L voor een Boolean vraag,

I voor een Integer vraag,

R voor een Real vraag en

T voor een Text vraag

ITEXT :de tekst van de vraag

De vragen moeten allemaal een verschillend, positief nummer hebben. De programmeur behoeft de "helptekst-links" niet zelf in te vullen; voornoemde utility "LINKCO" kan dit verzorgen.

Voorbeeld:

```
0 5Aantal kinderen
81 12Bgehuwd
264 7IDATUM
0 3Rwaarde
```

Vraag 5 luidt "aantal kinderen" en verwacht een integer antwoord, op vraag 12 ("gehuwd?") moet een Boolean (ja/nee) antwoord komen. Vraag 7 is die uit ons voorbeeld. Voor de vragen 7 en 12 zijn "help"-teksten aanwezig.

De help-sectie.

De "help"-sectie bevat "help"-teksten, waaraan de commando's en/of de vragen kunnen refereren.

De "help"-tekst voor een vraag begint met:

< vraagnummer

en een "help"-tekst voor een commando met:

< commando

Het "commando" mag beginnen met de hoofdnaam en kan subcommando's hebben. Het behoeft echter geen volledig commando te zijn. Alle namen mogen in de afgekorte vorm, zoals die in de commando-sectie gedefinieerd is, aangegeven worden. Het kleiner-teken "<" uit de "help"-sectie moet in de eerste kolom staan, in de "help"-tekst zelf mag dit teken niet in de eerste kolom voorkomen. Een "help"-tekst eindigt bij de indicatie van een andere "help"-tekst. De laatste "help"-tekst wordt afgesloten door "<<" of door "end-of-file" wanneer de implementatie daarin voorziet. In een "help"-tekst mogen regels opgenomen worden met in de eerste kolom een groter-teken ">". Bij het verstrekken van de tekst stopt COIN bij deze regel, meldt aan de gebruiker dat er nog meer informatie

is en wacht op een reactie. Stuurt de gebruiker een lege regel dan wordt de rest verstrekt, stuurt hij iets anders dan wordt de rest overgeslagen en de invoer in behandeling genomen.

Suggestie:

- Stuur niet meer dan ongeveer 20 regels tekst tegelijk; denk aan de gebruiker achter een beeldscherm.
- Probeer een wat langere "help"-tekst zo in te delen dat het eerste stuk min of meer complete informatie bevat en volgende stukken toelichtingen daarop zijn.

Voorbeeld van een helpsectie:

< OK SELECTEER

Met het SELECTEER commando kunt U een groep van personen uitzoeken.

U kunt dat doen op grond van hun namen, geboortedata of salaris. Zie daarvoor de betreffende subcommando's.

< 7

U moet een datum intikken in de vorm van een zes-cijferig getal. De eerste twee cijfers vormen het jaartal in deze eeuw, de volgende twee de maand (januari is 1, februari 2 etc.) en de laatste twee de dag-van-de-maand.

Voorbeeld: 770512 betekent 12 mei 1977

< OK

Druk op RETURN wanneer U niet weet wat te doen. Als U uitgebreidere hulp wenst gebruik dan het vraagteken op ieder gewenst moment. Ook:

? commandonaam

kan U gerichte informatie verschaffen.

>

Wanneer U een vraag gesteld krijgt staat er altijd een suggestie voor het antwoord bij. Bent U daar tevreden mee, dan hoeft U alleen maar op RETURN te drukken.

U mag altijd zoveel "vooruit" intikken als U wilt. Gebruik een of meer spaties of een komma als scheider wanneer dat nodig lijkt; tussen twee komma's mag een antwoord weggelaten worden; de suggestie, die bij het daar relevante commando hoort, wordt dan gebruikt. Wilt U alle volgende vragen onbeantwoord laten, gebruik dan de puntkomma; daarachter kunt U onmiddellijk het volgende commando zetten.

<<

Als een, door COIN ondersteund, programma door onervaren mensen gebruikt zal gaan worden dan blijkt de volgende inrichting plezierig:

- 1. Help de gebruiker op weg door bij de start van het programma enkele regels tekst te verstrekken, bijvoorbeeld de volgende:

Druk op RETURN als U niet weet wat te doen

Tik een vraagteken (?) voor uitgebreide hulp

- 2. Stel steeds algemene hulp beschikbaar. Deze mag best erg uitgebreid zijn; de gebruiker kiest immers zelf voor de mate waarin hij er gebruik van maakt.
- 3. Introduceer zonodig een HELP-commando om de betekenis en/of werking van Uw programma uit te leggen. Bovenstaande twee "binnenkomst"-regels kunnen dan uitgebreid worden met een derde:

Gebruik HELP voor inhoudelijk informatie

U kunt die "help"-tekst in de INFCO file opnemen als hulp bij HELP.

5. Ervaringen.

5.1. De gebruiker.

De afdeling der werktuigbouwkunde heeft op dit moment vier jaar ervaring in het gebruik van COIN. In eerste instantie werd dit pakket ingezet bij de ontwikkeling van onderwijsprogramma's. Studenten worden direct na de aanvang van hun studie in contact gebracht met een computerondersteunde vorm van onderwijs. De eerste confrontatie, die plaats vindt in de eerste week van hun verblijf in onze afdeling, is mede bedoeld om met terminals en onze gebruiksomgeving kennis te maken. In een pamflet wordt aangegeven hoe zij contact kunnen krijgen met het computersysteem. Het programma - spelen met eenvoudige voorwerpen in drie dimensies - wordt geactiveerd met een commando, dat er tevens voor zorgt dat men in de juiste gebruikersomgeving komt. Daarnaast wordt aangegeven dat men binnen de applicatie steeds een lijst van toegestane commando's kan krijgen door het drukken op de RETURN-toets en dat men aanwijzingen kan krijgen door HELP <commandonaam> in te tikken. Tenslotte wordt in enkele alinea's aangegeven wat het doel van de opdracht is en welke opdrachten men dient uit te voeren. De eerste opdracht is bedoeld om het perspectivisch inzicht te verdiepen. De studenten werken daarbij in groepjes van twee aan de opdracht.

Ondanks het feit dat veel studenten ook nog aan het werken met een terminal moeten wennen blijkt dat het "bedienen" van de applicatie in minder dan dertig minuten geleerd wordt. Men concentreert zich daarna op de opdrachten. Deze hebben een speleffect in zich, waardoor het totale leerrendement goed is. Nadien wordt er steeds vanuit gegaan dat onze wijze van communiceren bekend is; bij volgende onderwijsopdrachten wordt slechts een uiteenzetting van de opdracht zelf en de activering van de bijbehorende programmatuur gegeven. De instructeurs behoeven vrijwel nooit nadere uitleg over het gebruikersinterface te geven.

Alle serviceprogrammatuur binnen de afdeling wordt ook geconstrueerd met gebruikmaking van COIN. Ook voor de gebruikers van deze programmatuur geldt dat men zeer snel de gewenste interactie in de vingers heeft.

5.2. De programmeur.

Studenten, die in de tweede fase van hun studie zelf een programma moeten schrijven, maken vrijwel steeds automatisch gebruik van COIN. Dank zij het feit dat zij de uitwendige eigenschappen van COIN reeds kennen verloopt deze confrontatie erg soepel. Omdat men door het gebruik van COIN verlost is van het, met bijvoorbeeld FORTRAN read- en write-statements, construeren van steeds weer een vrijwel identieke communicatie en omdat men de mogelijkheid heeft om adequaat op onjuiste invoer te reageren zonder extra programmeerinspanning wordt erg enthousiast gereageerd op deze faciliteit.

Verder valt op dat de programma's veel gestructureerder van opzet worden. Omdat communicatie- en controle-algoritmen ontbreken worden programma's veel compacter en overzichtelijker. Doordat de afhandeling van bepaalde acties in afgepaste stukjes programma kan geschieden, krijgt een programma in FORTRAN vaak de vorm van een aantal geneste "Computed GO TO"-statements, waarbij de "transfer-control" geschiedt op basis van een door KEYCO afgeleverde KEY.

Voorbeeld FORTRAN IV:

```

(head)      CALL KEYCO (key, masterkey)
            GO TO (k1, k2, ... ,kn, kterm) key
k1  - - - - -
            GO TO head
k2  - - - - -
            GO TO head
..      - - - - -
            GO TO head
kn  - - - - -
            GO TO head
(term)kterm CONTINUE

```

Binnen de omgeving die door een "labelled statement" afgeschermd wordt kan een vergelijkbare constructie gemaakt worden, waardoor de nesting ontstaat.

De betrouwbaarheid van de programma's wordt ook aanmerkelijk groter omdat gegarandeerd wordt dat de ingevoerde variabelen van het gewenste type zijn en omdat men de mogelijkheid heeft om het waardegebied van een invoervariabele aan te geven.

Ook de medewerkers maken veelvuldig gebruik van COIN en geven aan dat door het gebruik van pakketten als COIN het schrijven van een programma is geworden tot het beschrijven van het werktuigbouwkundige proces, waarmee men in het algemeen goed bekend is, en het inbedden van dit proces in een prettige omgeving. Daar binnen de afdeling der werktuigbouwkunde een, met COIN vergelijkbare, ontwikkeling heeft plaatsgevonden voor de grafische presentatie van gegevens, is aan de uitvoerkant een vergelijkbare strategie mogelijk.

Binnen de afdeling wordt gepropageerd om zo min mogelijk van specifieke taalextensie's van bepaalde fabrikanten gebruik te maken en indien dit noodzakelijk is, deze te concentreren in een beperkt aantal "systeemafhankelijke" subroutines.

COIN neemt "free-format" invoer, inclusief de foutafhandeling en een groot gedeelte van de tekstmanipulatie voor zijn rekening. Met name niet professionele programmeurs waarderen dit in hoge mate. Omdat COIN op alle daarvoor in aanmerking komende systemen geïmplementeerd is, wordt een gebruikersprogramma erg transportabel. FORTRAN-programma's kunnen bijvoorbeeld in onze afdeling onder bovenstaande voorwaarden met zeer geringe aanpassing verwerkt worden op BURROUGHS B7900, PRIME, VAX, PDP, IBM-PC's en onder UNIX.

6. Verdere ontwikkelingen.

Zoals in het vorige hoofdstuk reeds aangegeven is, wordt binnen de afdeling der werktuigbouwkunde enthousiast gebruik gemaakt van deze commando-interpretator. De ontwikkelingen die nu plaats vinden hebben daarom geen conceptuele gevolgen. Zij zijn te zien als opties op het in het voorgaande beschreven basisconcept.

Daarnaast vinden er een aantal ontwikkelingen plaats omdat het toepassingsgebied zich uitbreidt. De structuur en de in deze structuur gebruikte routines blijken goed bruikbaar te zijn bij de ontwikkeling van andere applicaties.

Enkele ontwikkelingen:

- Alternatieve prompting.
In de prompt wordt historische informatie opgeslagen in die zin dat als prompt het gevolgde pad gegeven wordt.
- Uitbreiding "treewalk"-faciliteiten.
Er worden hierbij een aantal speciale symbolen gereserveerd om terug te gaan via het gevolgde pad. Gedacht wordt aan "/" om terug te gaan naar het hoofdniveau en "!" om één niveau omhoog te gaan in de commandoboom.
- Sturing van het niveau, waarop men terecht komt na afhandeling van een compleet commando.
- Globale menu's.
Indien men programmastructuren ontwikkelt waarbij een hoofdprogramma op basis van de gewenste applicatie zelfstandig andere programma's aanroept, dan is het in veel gevallen relevant dat het menu van het hoofdprogramma actueel blijft. Ook standaardcontacten met het operatingsysteem, zoals het opvragen van de gebruikte CPU-tijd en boodschappen naar andere gebruikers vallen hieronder. Dergelijke constructie's zijn mogelijk als "super-menu's" gedefinieerd kunnen worden.
- Het linken van commandostructuren.
Het komt vaak voor dat bepaalde sub-commando's op meerdere plaatsen relevant zijn. Dit kan men bereiken door op elke plaats waar dit nodig is, deze substructuur in te vullen. Een andere oplossing is de onderhavige subcommando's in een lijst op te nemen. Daar waar nodig wordt in dat geval naar deze lijst gerefereerd.
- Het gebruik van zelf gedefinieerde afkortingen.
Dit onderwerp heeft twee aspecten. Op de eerste plaats kan men dit gebruiken om volledige commando's of gedeelten daarvan te voorzien van een kenmerkende naam. Het ingegeven van deze naam heeft dan het zelfde effect als het (sub-)commando. Daarnaast

kan men de gebruiker in de gelegenheid stellen om commando's van een eigen naam of een eigen afkorting te voorzien.

- Recovery.

Het opslaan van het totaal van actie's, waardoor het mogelijk wordt om een bepaald handelingenpatroon opnieuw "af te spelen". Bij demonstratie's kan men ook van een dergelijk mechanisme gebruik maken.

- Uitvoer.

Ofschoon dit duidelijk buiten het destijds geformuleerde toepassingsgebied valt, kan men zich voorstellen dat vergelijkbare faciliteiten geboden worden ten behoeve van de uitvoer. Men denkt aan de ontwikkeling van routines ten behoeve van de geformateerde uitvoer, compatibel met COIN doch als zelfstandige applicatie inzetbaar. Hierdoor krijgt een gebruiker de mogelijkheid om onafhankelijk van een programmeertaal zijn uitvoer te definiëren.

- Taalafhankelijkheid.

De taal, waarin COIN met de gebruiker communiceert, is een aspect dat intern afgehandeld wordt. Daarentegen is de taal, waarin de applicatie met de gebruiker communiceert, extern gedefinieerd. Door beide aspecten extern af te handelen en de mogelijkheid in te bouwen om dynamisch van taal te wisselen, wordt het geheel flexibeler.

In toenemende mate wordt COIN gebruikt voor de constructie van programmatuur om invoer voor grote, gecompliceerde rekenprogramma's samen te stellen. Het is relatief eenvoudig om met behulp van dit gereedschap programma's te maken die, semantisch en syntactisch, correcte invoer voor dergelijke programma's construeren. Men heeft dan tevens de mogelijkheid om de informatie, die anders in handleidingen opgezocht moet worden, in de vorm van HELP-functie's aan te bieden. Daarnaast kan men met COIN ook invoerinterpretators bouwen, die de invoer op een wijze "scannen"

die vergelijkbaar is met die van de hoofdapplicatie. Dit is met name van belang omdat invoerfouten bij veel applicatieprogramma's pas geconstateerd worden in een stadium waarin de verwerking reeds ver gevorderd is.

7.Literatuur.

- [1] Wolfgang Dzida: Das IFIB-Modell für Benutzerschnittstellen. Office Management, vol 31. 1983.
- [2] L.H.Th.M. van Beukering: Software-ergonomie. Ingenieursinformatie, 24 augustus 1984.
- [3] S.K. Card, T. P. Moran, E. Newell: The Psychology of Human-Computer Interaction. Lawrence Erlbaum, London 1983.
- [4] J.P.A. Banens: COIN gebruikershandleiding, Rekencentrum Technische Hogeschool Eindhoven (N.L.), oktober 1981.
- [5] DIN 66 234 en voorontwerpen daarvan.
- [6] Ingbert Kupka, Susanne Maass, Horst Oberquelle: Kommunikation in Mensch-Rechner-Dialogen. Proc GI -12 Jahrestagung (DAC-73-GIA, BSR) Berlin, Springer, 1982.
- [7] H.W.Hofs, J.Fadegon, H.J.Willemsse: Mens-Computer-Interactie en technieken voor het beschrijven van dialogen. Informatie jr 26 nr 10, oktober 1984.