

10 Informatiezoekend gedrag van software-gebruikers en de consequenties daarvan voor inhoud en vorm van een software-handleiding

M.F. Steehouder

Informatiezoekend gedrag

Software-handleidingen, zo betoogde ik tijdens het vorige VIOT-congres (Steehouder 1987), hebben in grote lijnen twee functies: een leerfunctie en een naslagfunctie. De eerste functie houdt in dat de gebruiker van een softwareprogramma met behulp van de handleiding moet *leren* werken met het programma. Bij de naslagfunctie gaat het erom dat de gebruiker via de handleiding hulp moet krijgen wanneer hij om een of andere reden vast komt te zitten. Over deze functie gaat het in deze bijdrage.

Wanneer we willen weten welke informatie een naslag-handleiding moet bevatten en op welke manier die het best aangeboden kan worden, dan is het nuttig als we het een en ander weten van de manier waarop mensen met zo'n handleiding omgaan. Dat is het aloude principe van de *publiekgerichtheid*.

Het lijkt verstandig om de vraag naar leesgedrag niet te beperken tot wat er gebeurt op het moment dat de gebruiker de handleiding al voor zich heeft liggen. Dan dreigt het gevaar dat we de vraag naar inhoud en presentatie van een handleiding beperken tot wat traditioneel de *leesbaarheid* wordt genoemd: het gemak waarmee de tekst wordt geïnterpreteerd. Onderzoek in de laatste tien jaar heeft laten zien dat het, zeker met het oog op praktische verbeteringen in de communicatie, belangrijk is om uit te gaan van het bredere concept *bruikbaarheid*. Dat concept vereist dat we ons verdiepen in doelen van de gebruiker en relevante omgevingsfactoren, en dat we die betrekken in ons onderzoek naar de kwaliteit van teksten.

De noodzaak van een verbreding zoals ik die hier bepleit, wordt geïllustreerd door een onderzoek dat beschreven wordt door Kern (1985). Het ging daar om handleidingen die bedoeld waren voor een garagebedrijf, en waarvan bekend was dat ze slecht functioneerden. Besloten werd om de handleidingen leesbaarder te maken aan de hand van de conventionele schrijfadviesen voor begrijpelijke teksten: er werd een duidelijke structuur in aangebracht, er werden eenvoudige woorden en zinnen gehanteerd, er werden duidelijke illustraties gegeven, enzovoort. Het resultaat beviel prima: de nieuwe handleiding werd zowel door deskundigen als door de gebruikers positief beoordeeld. Helaas bleek bij een evaluatie-onderzoek dat ook de nieuwe handleiding nauwelijks werd gebruikt. Pas bij een observatie van de monteurs in het betreffende bedrijf bleek dat hun *informatiezoekend gedrag* heel anders was, dan voorheen werd verondersteld. Zo bleken ze vaak niet te beseffen dat ze de problemen die ze tegenkwamen, beter zouden kunnen oplossen als ze de handleiding erbij zouden nemen. Vaak kozen ze ook voor een andere informatiebron, vooral de

Op grond van deze observaties, en als basis voor verder onderzoek, presenteert Kern (1985) een *model voor informatietoekend gedrag* dat in een ietwat verstrakte vorm is weergegeven in figuur 1.

INFORMATIEZOEKEND GEDRAG	
1	Bewustwording van informatie-behoefte
2	Formulering van de informatie-vraag
3	Selectie van de informatiebron
4	Zoeken van de relevante informatie
5	Interpretatie van de informatie
6	Evaluatie

Figuur 1 Model van informatietoekend gedrag (naar Kern 1985)

Het model is zo algemeen, dat een brede toepassing voor de hand ligt. Het lijkt in ieder geval ook geschikt om het informatietoekend gedrag van software-gebruikers te beschrijven. Maar het is tegelijk een tamelijk globaal model. Het geeft een zekere *fasering* van informatietoekend gedrag, maar het geeft niet aan hoe het proces verloopt. Het vraagt om verdere onderscheidingen en detaillering voordat het echt mogelijk is om toetsbare hypothesen op te stellen en/of bruikbare adviezen voor software-handleidingen te formuleren en te funderen.

In de beschikbare literatuur en op grond van enkele pilot-onderzoekjes zijn al wel aanzetten tot verdere detaillering te geven. Uit interviews met software-gebruikers blijkt bijvoorbeeld dat in de derde fase, de selectie van de informatiebron, collega's de populairste informatiebron vormen, gevolgd door de online-hulpfunctie, terwijl de (schriftelijke) handleiding alleen wordt geraadpleegd als men niet anders meer kan. Ook is er het nodige onderzoek gedaan naar de voordelen die typografische signalen, kopjes, marginalia, inhoudsopgaven en registers kunnen bieden bij de vierde fase, het opzoeken van relevante passages in teksten (zie de overzichtsartikelen in Jonassen 1985, p. 187-286). Voor de vijfde fase, de interpretatie, lijkt het onderscheid van belang tussen *semantische* en *pragmatische* interpretatie (Jansen & Steehouder 1989). Met het eerste wordt bedoeld dat de gebruiker van een handleiding moet begrijpen wat de tekst betekent, met het tweede dat hij moet afleiden hoe hij zijn probleem moet oplossen (een *handlingsplan* opstellen). Uit het genoemde onderzoek blijkt dat met name de laatste vorm van interpretatie bij het invullen van formulieren nogal eens voor problemen zorgt.

In deze beschouwing wil ik proberen de eerste twee fasen uit het model nader te verkennen, en dan speciaal in relatie tot software-handleidingen.

Bewustwording van informatiebehoefte

Wanneer en hoe ontstaat in een 'vertrouwde technische omgeving' zoals de situatie van de niet-meer-beginnende software-gebruiker gekarakteriseerd kan worden, de behoefte aan informatie die uiteindelijk kan leiden tot het raadplegen van een handleiding? Een eerste invalshoek biedt het theoretisch kader van Berlyne (1965), dat enkele jaren geleden enigszins geactualiseerd is door Inagaki & Hatanō (1986). In deze theorie worden drie soorten aanleidingen voor informatiebehoefte onderscheiden:

- *Surprise*: iemand wordt geconfronteerd met een gebeurtenis of met informatie die op een of andere manier conflicteert met zijn verwachtingen. Dat gebeurt bijvoorbeeld wanneer iemand werkend aan een software-programma een fout maakt of wanneer in het systeem een storing optreedt.
 - *Perplexity*: de situatie waarin iemand meerdere wegen ziet om een bepaald doel te bereiken, maar niet kan uitmaken welk van die wegen het meest adequaat is. Tot deze categorie aanleidingen zou men ook de situatie kunnen rekenen waarin iemand *geen* wegen tot een bepaald doel ziet.
 - *Discoordination*: iemand wordt geconfronteerd met een gebrek aan cohesie tussen verschillende kenniselementen. Het gaat niet zozeer om tegenstrijdigheid, maar om het feit dat ze niet met elkaar te maken lijken te hebben.
- De drie hier onderscheiden situaties hoeven niet *automatisch* te leiden tot informatie-behoefte. Men kan besluiten de probleemsituatie te negeren en/of te laten voortbestaan. Men kan bijvoorbeeld genoegen nemen met een sub-optimaal resultaat, of gebrek aan inzicht accepteren (als de oplossing maar werkt). De kans dat men beslist om het cognitieve conflict op te lossen door informatie te zoeken is het grootst als men voldoende gemotiveerd is om een oplossing te vinden, als men vertrouwen heeft in de mogelijkheid om die oplossing inderdaad te vinden.

Van deze drie soorten aanleidingen tot informatietoekend gedrag, is in relatie tot technische omgevingen, vooral de eerste voorwerp van studie geweest. Er heeft betrekkelijk veel onderzoek en theorievorming plaatsgevonden op het gebied van *menselijk falen* (human error) en *diagnostisch gedrag*. Onderzoekresultaten op dit gebied lijken van belang voor vragen over de rol die informatie uit handleidingen en dergelijke kan spelen, ook al is nog allerminst duidelijk welke rol dat precies is, laat staan hoe een handleiding optimaal kan functioneren.

Een samenvattend model van het *menselijk falen in een vertrouwde technische omgeving* is ontwikkeld door Reason (1987) (zie figuur 2); het sluit onder andere sterk aan bij het theoretisch kader van Rasmussen (1983; 1986).

Naar de manier waarop mensen in probleemsituaties diagnoses stellen en remedies ontwikkelen, is de laatste jaren op enige schaal onderzoek gedaan. Een recent onderzoek is dat van Groenewegen (1990). Zij onderzocht de reacties van proefpersonen op onverwachte informatie in verschillende omstandigheden. In een eerste serie experimenten moesten studenten een onverwachte situatie uit het dagelijks leven verklaren. In een tweede reeks experimenten moesten leerling-automonteurs er via vragen of door eigen waarnemingen achter zien te komen waarom een auto niet wilde starten. De derde serie experimenten had plaats in een complexe professionele technische omgeving: operators in de proces-industrie moesten een diagnose stellen van bepaalde configuraties op controleschermen. Groenewegen onderscheidt twee mogelijke strategieën bij het stellen van diagnoses (zie figuur 3).

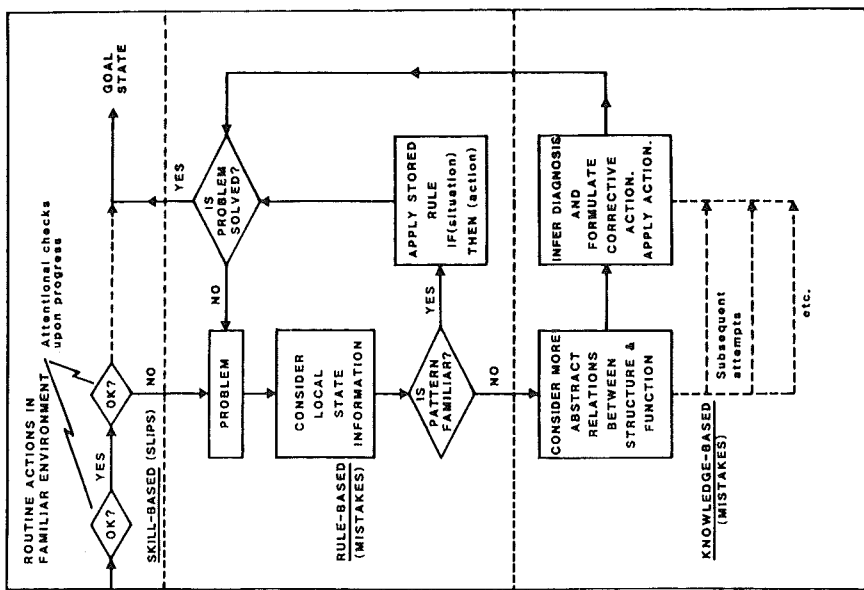
forward reasoning (voortredeneren)	backward reasoning (terigtredeneren)
<ul style="list-style-type: none"> - alle gegevens bekijken - mogelijke oorzaken inventariseren - alternatieven wegen - meest aannemelijke diagnose kiezen 	<ul style="list-style-type: none"> - snel, intuïtief hypothese opstellen - verificatie van de hypothese - als gefalsifeerd: nieuwe hypothese - na voldoende bevestiging: conclusie

Figuur 3 Twee strategieën bij het stellen van diagnoses

Alle drie de series experimenten suggereren dat het diagnoseproces wordt gekenmerkt door terugredeneren (backward reasoning). Men genereert op basis van enkele gegevens een hypothese, en test die vervolgens door naar de overige gegevens te kijken. Spreken die de hypothese niet tegen, dan worden ze beschouwd als bevestiging. Spreken ze de hypothese wel tegen, dan worden ze soms veronachtzaamd. De gegevens dienen behalve ter verificatie ook als middel om de diagnose verder te detailleren. Heeft de test naar het oordeel van de probleemoplosser voldoende zekerheid opgeleverd, dan wordt het proces gestaakt (dus niet alle noodzakelijke voorwaarden worden geverifieerd).

In dit diagnoseproces zijn vijf karakteristieke fouten te onderscheiden:

- 1 Fouten bij de selectie van de voor de diagnose relevante symptomen (men schat het belang van verschillende gegevens verkeerd in).
- 2 Onjuiste hypothese doordat niet alle relevante symptomen worden verdisconteerd.
- 3 Redeneerfouten bij de verificatie van de hypothese: bevestiging van de consequens en veronachtzaming van gegevens die de hypothese ontkrachten. Ook laat men vaak na de eerste hypothese te vergelijken met mogelijke alternatieven.
- 4 Fouten door te vroegtijdige beëindiging van het verificatieproces; vaak is men tevreden met een waarschijnlijk en/of sub-optimale oplossing.
- 5 Wanneer een hypothese verworpen wordt, wordt een nieuwe hypothese



Figuur 2 Algemeen model van het menselijk falen in een vertrouwde technische omgeving (uit Reason, 1987)

In het model van Reason wordt uitgegaan van een situatie waarin iemand min of meer routinematig aan het werk is, en plotseling ontdekt dat er iets verkeerd gaat. Er ontstaat een probleem. Is dat probleem op een of andere manier *vertrouwd*, dan kan gebruikt gemaakt worden van in het geheugen opgeslagen *regels* of procedurele kennis. Maar gaat het om een nieuw, niet eerder ontdekt probleem, dan moet er een *diagnose* gesteld worden en een *recept* geconstrueerd worden. Hiervoor, aldus Reason, is méér nodig dan kennis van regels of procedures; men moet inzicht hebben in de werking van het systeem. Let wel: kennis van regels en procedures is ook nu nodig. Maar er is kennis over de structuur van het systeem *als geheel* nodig om een kansrijke *selectie* en *combinatie* van de (deel-)procedures te kunnen maken en om het effect van bepaalde ingrepen te kunnen *evalueren*.

opgesteld zonder dat er enig logisch verband is met het tot dan toe verrichte werk.

De samenhang tussen de manier van redeneren en de verschillende fouten is te verklaren door de beperkingen van het menselijk brein: mensen kunnen niet alle aspecten in de redenering betrekken. In plaats daarvan zoeken ze naar overeenkomsten tussen de gegeven (nieuwe) situatie en situaties die in hun geheugen zijn opgeslagen: daar waar zich verschillen voordoen, wordt de diagnose van het probleem gezocht. Aangezien de nieuwe situatie en de geheugenrepresentaties slechts gedeeltelijk overeen kunnen komen, kan de diagnose nooit direct precies zijn. Naarmate bepaalde kenmerken sterker overeen komen of vaker voorkomen, zullen ze meer invloed hebben op de op te stellen hypothesen.

Een enigszins vergelijkbaar beeld komt naar voren in het onderzoek van Aaronson & Carroll (1987) naar de interactie tussen 'help desk consultants' en hun cliënten. Zij analyseerden dertig gesprekken en gingen onder andere na wat voor soort vragen hulpzoekers nu eigenlijk stellen. Uit de analyse bleek dat 76% van de vragen *verificatie-pogingen* waren: uitingen waarin een bepaald idee werd geopperd met de (expliciete of impliciete) vraag of dat correct was. Die ideeën konden zowel een diagnose van een probleem betreffen als een mogelijke oplossing. Bijvoorbeeld: *How do I make a data label - just get rid of the colon?* of: *If you make it NF Macro, I bet the CALL CUE goes away?* 12% van de gestelde vragen waren open vragen naar meer informatie en nog eens 12% betrof *rejections*: vragen waarin de aangeboden oplossing min of meer verworpen werd. Uiteraard uiten de cliënten ook 'acknowledgements': instemming met de aangeboden hulp. Zulke uitingen kwamen echter veel minder voor dan vragen: vragen kwamen 2,5 maal zoveel voor (maar instemming kan ook non-verbaal geuit worden).

Een nadere analyse van de verificatievragen wees uit dat er vier soorten te onderscheiden waren. In 8% van de gevallen ging het in wezen om een vraag naar meer informatie (*Is er een commando voor X?*). In 6% van de gevallen betrof het de herhaling van door de hulpverlener ingebrachte adviezen of stappen. In 23% was er sprake van een parafraze, eventueel aangevuld met een triviaal detail van eerder door de hulpverlener gegevens adviezen of stappen. In 63% van de gevallen betrof het 'echte' nieuwe ideeën (hypothesen). Het percentage nieuwe ideeën bleek bij ervaren gebruikers groter (72%) dan bij minder ervaren gebruikers (33%).

Zowel het theoretisch kader van Reason als de onderzoeksresultaten van Groenewegen en van Aaronson & Carroll geven grond aan de veronderstelling dat mensen in *surprise*-achtige situaties niet zozeer een beroep doen op kennis van regels, maar dat ze *hypothetische* diagnoses en oplossingen bedenken, en die vervolgens toetsen. Zulk gedrag doet een beroep op méér dan procedurele kennis: men moet de structuur van het systeem doorzien, relevante kenmerken identificeren, en op basis van dit inzicht een oplossing bedenken. Groenewegen betoogt dat de huidige hulpsystemen in technische omgevingen (handleidingen, online hulp e.d.) vooral het gedrag op regulier niveau ondersteunen. De onderzoeksresultaten wijzen op het belang van de ontwikkeling van systemen die op kennisniveau hulp bieden door de selectie van relevante symptomen en de verificatie door middel van gerichte vragen te ondersteunen.

Consequenties voor handleidingen

Een weinig omstreden gezichtspunt omtrent instructieve teksten zoals handleidingen is, dat de inhoud bepaald moet worden door de *handelingen van de gebruiker*. In verschillende publicaties wordt dit inzicht naar voren gebracht met etiketten als *performance oriented writing* (Kern e.a. 1975), *Scenario-principle* (Flower e.a. 1983) of *handelingsperspectief* (Jansen & Steehouder 1989). De consequenties van deze perspectiefkeuze voor de tekst zijn in enkele steekwoorden samen te vatten (zie figuur 4).

<i>topic-oriented declarative conceptueel perspectief</i>	<i>performance-oriented procedureel handelingsperspectief scenario-principle</i>
structuur van het systeem	handelingsverloop van de gebruiker
beschrijving	instructie
logische ordening	chronologische ordening
gericht op kennis (inzicht)	gericht op toepassing (vaardigheid)

Figuur 4 Perspectiefkeuze in een instructieve tekst

De empirische fundering voor de keuze van het handelingsperspectief is zeer beperkt. Kern (1985) wijst vooral op de 'logica' van een dergelijke keuze voor teksten die primair bedoeld zijn om het *gedrag* van mensen te sturen (teksten die bedoeld zijn voor wat Sticht (1985) karakteriseert als *reading to do*). Het *scenario principle* van Flower e.a. (1983) wordt ondersteund door een experiment waarin bleek dat lezers van overheidsregelingen de beschrijvende informatie uit de tekst vaak omzetten in scenario's: beschrijvingen van menselijke acties in bepaalde situaties.

De consequenties van deze perspectiefkeuze demonstreer ik aan de hand van een voorbeeld: een passage uit een (niet geheel denkbeeldige) handleiding voor een tekstverwerkingsprogramma (figuur 5). De hier gegeven instructie volgt weliswaar in eerste instantie de handelingen die de gebruiker moet uitvoeren, maar rechtsonder in de instructie blijkt dat er een adder in het gras zit. Het daar genoemde *stoptekens* staat natuurlijk nooit toevallig in een tekst; dat heeft de gebruiker er om een bepaalde reden ingezet. De geschreven instructie in figuur 6 lijkt daarom duidelijker: die geeft preciezer aan welk handelingsverloop de gebruiker onder welke condities moet volgen.

actually doing anything except following these directions.

Zoals ik in het eerste deel van het artikel betoogde, kan de vraag naar de effectiviteit van de instructie niet los gezien worden van de aanleiding om de instructie op te zoeken. Gebruikers die (nog) niet weten hoe ze een tekstpassage moeten verwijderen, zullen ondanks het eventuele gebrek aan inzicht geholpen zijn met de instructie uit figuur 6. Voor gebruikers die naar aanleiding van een *surprise* bij de betreffende passage terecht komen, kan dat wel eens anders zijn. Zo'n *surprise* kan bijvoorbeeld ontstaan op het moment dat men (per ongeluk) teveel tekst verwijdert: men verwijdert bijvoorbeeld drie volle regels, terwijl het de bedoeling was om van de derde regel alleen de eerste vijf woorden te verwijderen.

Zowel een diagnose als een remedie lijkt in zo'n situatie alleen mogelijk als de gebruiker inzicht heeft in de *systematiek* van het software-programma, en speciaal in de functie van het stopteken. Dat inzicht kan bijvoorbeeld gegeven worden in een passage als figuur 7. Ook andere toelichtingen zijn mogelijk, maar het ligt toch in de rede dat die min of meer *beschrijvend* zijn, en niet strikt volgens het handelingsprincipe zijn opgesteld.

Hoe verwijdert u een passage uit een tekst?	
Wat te doen?	Wat gebeurt er?
1 Zet de cursor aan het begin van de passage die u wilt verwijderen.	
2 Geef het commando Kr , waarbij <i>n</i> staat voor het aantal regels dat u wilt verwijderen.	De passage verdwijnt van het scherm. Als er een <i>Stopteken</i> (■) in de passage staat, wordt de tekst daarachter niet verwijderd.

Figuur 5 Voorbeeld: passage uit een computerhandleiding

Hoe verwijdert u een passage uit een tekst?		
Als de passage...	Ga dan als volgt te werk:	Wat er gebeurt:
aan het eind van een regel eindigt...	<ol style="list-style-type: none"> Verwijder zonnodig alle stoptekens in de passage. Zet de cursor aan het begin van de passage die u wilt verwijderen. Geef het commando Kr, waarbij <i>n</i> staat voor het aantal regels dat u wilt verwijderen. 	De passage verdwijnt van het scherm.
midden in een regel eindigt...	<ol style="list-style-type: none"> Zet een stopteken (<i>Alt-F9</i>) op de plaats tot waar u de tekst wilt verwijderen. Verwijder zonnodig alle andere stoptekens in de passage. Zet de cursor aan het begin van de passage die u wilt verwijderen. Geef het commando Kr, waarbij <i>n</i> staat voor het aantal regels dat u wilt verwijderen (inclusief die met het stopteken). 	U ziet een <i>Stopteken</i> De passage verdwijnt van het scherm.

Figuur 6 Verder uitgewerkte instructie (vergelijk figuur 5)

Heeft de consequente toepassing van het handelingsperspectief nu tot een duidelijker instructie geleid? Dat valt nog te betwijfelen. Het handelingsverloop is aanzienlijk complexer, en het risico is niet denkbeeldig dat er op zijn minst *discoördinatie* ontstaat. Tekenend is de observatie van Mack e.a. (1983) dat beginnende softwaregebruikers een duidelijke weerstand hebben tegen een overdaad aan stap-voor-stap procedures. Dat leidde tot uitspraken van hardopdenkend proefpersonen als *I know I did something but I don't know what it is* of *I'm getting confused because I'm not*

Het stopteken

In **WPLUS** kan een document een zogenaamd stopteken bevatten. Op het scherm is dat zichtbaar als een zwart vierkantje (■). Verschillende commando's gaan niet verder dan dit stopteken. Dat is het geval bij de volgende commando's:

- *printen* (*P*): de tekst na te stopteken wordt niet geprint.
- *kladgebied* (*K*): de tekst na het stopteken wordt niet verplaatst naar het kladgebied.
- *wissen* (*W*): de tekst na het stopteken wordt niet gewist.
- *schrijven* (*S*): de tekst na het stopteken wordt niet naar diskette geschreven.

Het gaat dus steeds om commando's waarbij een aantal regels opgegeven moet worden.

Figuur 7 Toelichting op de structuur van een programma (vergelijk figuur 5 en 6)

In zijn algemeenheid lijkt het plausibel dat het handelingsperspectief in veel situaties een goed uitgangspunt is, maar dat het niet tot dogma verheven kan worden. Vooral wanneer de gebruiker voor problemen komt te staan die niet opgelost kunnen worden via vaste, min of meer routinematige regels, lijkt een andere benadering geboden. Groenewegen (1990) stelt, waarschijnlijk terecht, dat de meeste technische hulpsystemen (handleidingen, online hulp e.d.) vooral het gedrag op regelniveau ondersteunen. De literatuur die ik in het eerste deel van dit artikel heb samengevat en de observaties in het tweede deel lijken erop te wijzen dat voor problemen op kennisniveau een andere benadering nodig is. De kennis moet daar worden aangeboden op een wijze de diagnose van een probleem ondersteunt en die het de gebruiker mogelijk maakt een 'eigen' oplossingsmethode te construeren.

Op welke wijze dit doel optimaal valt te realiseren, is voorslagnog niet erg helder. In eerste instantie kan gedacht worden aan verschillende technieken, zoals schema's, analogieën, en overzichten. Een complicatie daarbij is dat gebruikers van handleidingen dikwijls weinig belangstelling lijken te hebben voor zulke *elaboraties*: ze zoeken zo snel mogelijk de aanwijzingen die ze in de gegeven situatie nodig hebben. De rest is *maar informatie* en wordt bij voorkeur niet gelezen (zie bijvoorbeeld Wright 1989 en de vergelijkbare houding van formuliereninvullers in Jansen & Steehouder 1989).

Literatuur

- Aaronson, A., & Carroll, J.M.
1987 The answer is in the question: a protocol study of intelligent help. In: *Behaviour and information technology* 6, p. 393-402.
- Berlyne, D.E.
1965 *Structure and direction in thinking*. New York.
- Flower, L., J. Hayes & H. Swarts
1983 Revising functional documents: the scenario principle. In: P.V. Anderson, R.J. Brockmann & C.R. Miller (eds.): *New essays in technical and scientific communication. Research, theory and practice*. Farmingdale, p.41-58.
- Groenewegen, A.J.M.
1990 *What Happened? Diagnosing unfamiliar real-life situations*. Leiden.
- Inagaki, K., & G. Hatano
1986 *Motivation for understanding in the classroom: a cognitive Berynean theory*. Unpublished paper, Chiba University & Dokkyo University (JP).
- Jansen, C.J.M., & M.F. Steehouder
1989 *Taalverkeersproblemen tussen overheid en burger. Een onderzoek naar verbeteringsmogelijkheden van voorlichtings teksten en formulieren*. 's-Gravenhage.
- Jonassen, D.H. (ed.)
1985 *The technology of text. Principles for structuring, designing and displaying text*. Vol. 2. Englewood Cliffs NJ.
- Kern, R.P.
1985 Modeling users and their use of technical materials. In: T.M. Duffy & R.H. Waller (eds.): *Designing usable texts*. New York etc., p.341-375.
- Kern, R.P., T.G. Sticht, D. Welty & R.N. Hauke
1975 *Guidebook for the development of army training literature*. Human Resources Research Organization, Alexandria VA.
- Mack, R.L., C.H. Lewis & J.M. Carroll
1983 Learning to use word processors: problems and prospects. In: *ACM Transactions on Office Information Systems* 1, p. 254-271.
- Rasmussen, J.
1983 Skills, rules, and knowledge; signals and symbols, and other distinctions in human performance models. In: *IEEE transactions on systems, man, and cybernetics*, 5, p. 257-266.
- Rasmussen, J.
1986 *Information processing an human-machine interaction. An approach to cognitive engineering*. New York etc. North-Holland series in system science and engineering, Vol. 12.
- Reason, J.
1987 Generic error-modelling system (GEMS): a cognitive framework for locating common human error forms. In: J. Rasmussen, K. Duncan & J. Leplat (eds): *New technology and human error*. London, p. 63-83.
- Steehouder, M.
1988 Hoe gaan computergebruikers met een handleiding om?. In: F. van Eemeren & R. Gooten-dorst (red.): *Taalbeheersing in ontwikkeling. Lezingen van het VIOT-taalbeheersingscongres gehouden op 16, 17 en 18 december 1987 aan de Universiteit van Amsterdam*. Dordrecht. p.265-272.
- Sticht, T.
1985 Understanding users and their use of text. In: T.M. Duffy & R.H. Waller (eds.): *Designing*

usable texts, New York etc., p.315-340.

Wright, P.

1989 The need for theories of NOT reading: some psychological aspects of the human-computer interface. In: I.A.G. Elsendoorn & H. Bouma (eds): *Working models of human perception*. London etc, p. 319-340.