Kantoorautomatisering en verder..... 15

Tekstverwerking: De vijfde generatie

Prof. dr. Gerard Kempen

In de jaren tachtig hebben de eerste commerciële toepassingen van Artificiële Intelligentie (AI) het licht gezien. De twee belangrijkste voorbeelden zijn expertsystemen en natuurlijke-taalverwerkers. Deze ontwikkelingen beginnen hun stempel te drukken op het ontwerp van kantoormachines. Toekomstige kantoor-werkstations zullen enig verstand hebben van kantoorprocedures en kunnen 'meedenken' met degenen die ze uitvoeren. Ook zullen deze stations voorzien zijn van een natuurlijke-taaldialoogsysteem. Dit zal kantoorfunctionarissen in staat stellen om met het werkstation te communiceren in gewone taal. Bovendien zullen werkstations als een soort taalconsulent diensten kunnen verlenen bij het schrijven en vertalen van teksten.

1 De huidige situatie

De volledige realisatie van dergelijke functies in betaalbare kantoorcomputers laat zeker nog jaren op zich wachten. Niettemin zijn aanzienlijke onderdelen ervan nu reeds binnen het bereik van krachtige microcomputers: eenvoudige Engelstalige vraag-antwoordsystemen, semi-automatische vertaalsystemen, en linguïstische editors. Met deze laatste bedoel ik speciale tekstverwerkingsprogrammatuur die auteurs en typisten helpt bij het controleren van spelling, zinsbouw en stijl. Goede voorbeelden zijn de Writer's Workbench die is ontwikkeld op Bell Labs en IBM's CRITIQUE (vroeger EPISTLE geheten). Diverse op micro-computers draaiende systemen voor semi-automatisch vertalen zijn momenteel op de markt. Ze bieden een op vertaalwerk afgestemde tekstverwerkingsomgeving en kunnen een eerste ruwe vertaling afleveren die vervolgens door de vertaler wordt gecorrigeerd. Sommige van deze microcomputers kunnen als werkstation aangesloten worden op meer geavanceerde vertaalsystemen die op grotere computers draaien.

Rapporten over kosten-baten-analyses van machinale vertaling zijn inmiddels verschenen. In 1983 werd de eerste Conference on Applied Natural Language Processing georganiseerd door de Amerikaanse Association for Computational Linguistics (ACL). Daar presenteerde Jonathan Slocum de uitkomsten van een vergelijkende studie waarin de normale vertaalprocedure (vertaling door een menselijke vertaler en correctie achteraf door een collega) afgezet werd tegen machinevertaling (waarbij de machinaal geproduceerde tekst door een vertaler werd gereviseerd). De totale kosten van semi-automatische vertaling, met inbegrip van apparatuur en programmatuur, bleken lager uit te vallen dan die van de normale procedure. De vertalingen die in de twee condities tot stand kwamen, werden beoordeeld als vergelijkbaar in kwaliteit. Een belangrijk praktisch voordeel van computerondersteunde vertaling is de verkorte looptijd van vertaalprojecten.

2 Dialoogsystemen

Een handig schema om de kwaliteit van dialoogsystemen te beoordelen is enkele jaren geleden opgesteld onder auspiciën van de ACL (Hendrix, 1982). Drie niveaus van taalvaardigheden worden onderscheiden waarover dialoogsystemen kunnen beschikken. Op het laagste niveau (niveau 1) bewegen zich alle systemen die momenteel geschikt zijn voor praktisch gebruik. Tot deze categorie behoren de weinige systemen die als commerciële softwarepakketten verkrijgbaar zijn (INTELLECT, Q&A). Systemen van niveau 2 bestaan nog slechts als prototypes in laboratoria, en het werk aan niveau-3-systemen is net begonnen.

De volgende globale karakterisering van de drie niveaus kan dienen als vertrekpunt. Niveau-1-systemen (zie ook Tabel 1) bieden een basispakket van voorzieningen die in elk dialoogsysteem thuishoren, dat wil zeggen, modules
- voor de ontleding en interpretatie van veelvoorkomende woorden, woordgroepen en zinnen (met name vraagzinnen);
- voor het genereren van antwoordzinnen;
- voor uitbreiding en aanpassing van het lexicon dat de woorden en uitdrukkingen bevat waar vragensteller en antwoordgenerator uit kunnen putten;
- voor het construeren van programma's in de formele taal (commandotaal, zoektaal, stuurtaal) waarmee de databank werkt.

Niveau 2 kent krachtige methoden voor het oplossen van referentiële beschrijvingen (dat wil zeggen diverse typen van naamwoordsgroepen die verwijzen naar objecten, personen, toestanden, gebeurtenissen en dergelijke in het domein waarop de databank betrekking heeft). Het ontwerp van niveau-2-dialoogsystemen belichaamt een aantal nieuwe principes. De belangrijkste zijn:
- de linguïstische modules laten zich in sterke mate leiden door 'kennis van de wereld' (dat wil zeggen door een representatie van het inhoudelijke domein van de databank);
- de gesprekscontext (dat wil zeggen de voorafgaande vragen, opdrachten, antwoorden en dergelijke alsmede hun betekenissen) blijft bewaard en wordt volledig benut;
- invoerzinnen worden niet rechtstreeks in de formele taal van het informatiesysteem omgezet maar in

INFORMATIE JAARGANG 29 NR. 5
PAG. 397 T/M 492 1987

een logische tussentaal, vaak een vorm van eerste-orde predicatenlogica. (Allerlei taalverschijnselen laten zich nu op algemenere en linguïstisch meer adequate wijze behandelen, zodat een groot aantal ad hoc oplossingen overboord gezet kan worden; en dit terwijl de vertaalslag van logische tussentaal naar formele (commando-) taal betrekkelijk eenvoudig is.)

Deze nieuwe basisprincipes stellen dialoogsystemen in staat tot aanzienlijk betere prestaties bij het interpreteren van referentiële beschrijvingen dan hun voorgangers. De volgende voorbeelden laten hier iets van zien. Stel, een informatiesysteem over autoreparaties wordt gevraagd: 'Als ik koelwater wil bijvullen, waar vind ik dan de dop?'. De beslissing dat hier de dop van de radiateur en niet van bijvoorbeeld de benzinetank bedoeld moet zijn, berust niet op linguïstische regels maar op kennis van autotechniek.
De gesprekscontext kan evenmin buiten beschouwing blijven bij het oplossen van referentiële beschrijvingen. Neem het voorraadadministratiesysteem van een boekhandel dat de opdracht krijgt: 'Geef voorradige titels en prijzen'. Als het dialoogsysteem op niveau 1 functioneert zal het een volledige lijst van alles wat in voorraad is op het scherm brengen. Een dialoogsysteem van niveau 2 houdt echter rekening met de inhoud van het voorafgaande. Bijvoorbeeld na 'Ken je boeken van Marga Minco?' zal de uitdraai van titels en prijzen beperkt blijven tot haar werk. Ook de naamwoordgroep 'hetzelfde' in 'En nu hetzelfde voor Doeschka Meijsing' wordt correct geïnterpreteerd. (Dit is tevens een voorbeeld van pronominalisatie waar niveau-1-systemen niet uitkomen: 'hetzelfde' slaat niet terug op een eerder genoemd object maar op een voorafgaande zin uit de dialoog.)
Een dialoogsysteem van niveau 3 zal kunnen putten uit een belangrijke extra bron van informatie: kennis over doelgericht gedrag. Uitgaande van behoeften en doelstellingen die het aan de gebruiker, aan zichzelf en eventueel aan derden toekent, zal het plannen kunnen ontwikkelen en voorstellen die zo goed mogelijk aan de belangen van deze partijen tegemoet komen. De antwoorden van het dialoogsysteem zijn op zulke plannen gebaseerd. Wilensky (1982) is als een van de eersten aan de bouw van zo'n systeem begonnen. Hij werkt aan een 'consulent' die aan gebruikers van het bekende bedrijfssysteem UNIX adviezen geeft wanneer ze in problemen geraken. De consulent is met name bedoeld om onervaren gebruikers op weg te helpen. De vraag-antwoordparen in tabel 2 illustreren dit.

Deze paragraaf heeft een opklimmende reeks taalvaardigheden waarover dialoogsystemen nu of binnen afzienbare tijd kunnen beschikken, de revue laten passeren. Misschien is de indruk gewekt dat we over enkele jaren met computers kunnen communiceren net zoals met mensen. Niets is echter minder waar. De laatste tijd zijn de concepties die aan het ontwerp van dialoogsystemen ten grondslag liggen, sterk verbeterd, en deze 'verticale' trend van kennis*verdieping* zal zich ook in de toekomst voortzetten. Dit leidt echter niet vanzelf tot *verbreding* van kennis en vaardigheid binnen niveaus. Hedendaagse dialoogsystemen opereren altijd op een zeer smalle kennisbasis. En uitbreiding in de breedte is allerminst triviaal. Het is arbeidsintensief omdat grote kennishoeveelheden geïnventariseerd, geanalyseerd en foutloos gecodeerd moeten worden. Bovendien legt het een groot beslag op extern geheugen zodat responsietijden schrikbarend kunnen stijgen.
Deze beperkingen gelden evenzeer voor andere taaltechnologische toepassingen. Volledig automatisch vertalen bijvoorbeeld is onmogelijk zonder goed begrip van de inhoud van de brontekst. Dit vergt een grote hoeveelheid domeinkennis.

3 Auteursomgevingen

Ik stel de term *auteursomgeving* of *auteursysteem* voor als benaming voor alle programmatuurlijke gereedschappen die auteurs (en redacteurs) ten dienste kunnen staan bij het schrijven, bewerken en afhandelen van teksten die in een natuurlijke taal zijn gesteld.
Het is een belangrijke tekortkoming van hedendaagse tekstverwerkers dat ze vrijwel verstoken zijn van taalkennis. Deze blijft beperkt tot (gebrekkige) regels betreffende spelling en woordafbreking. In deze situatie zal evenwel verbetering komen. De eerste tekenen daarvan zijn de al genoemde Writer's Workbench en CRITIQUE. Auteurs die in het Engels schrijven kunnen met behulp van deze gereedschappen hun spelling, zinsbouw en stijl nakijken. Niet alleen spelling en interpunctie worden gecontroleerd maar ook aspecten van zinsbouw en stijl: lange zinnen, abstracte woordkeus, ontbrekende congruentie van onderwerp met persoonsvorm, en andere schendingen van grammaticaregels.
De diagnose die zulke programma's stellen zijn beslist niet nauwkeurig en waterdicht. Ze zijn immers gebaseerd op onvolledige taalkundige analyse van de ingevoerde tekst. Krachtiger technieken voor het linguïstisch ontleden en inhoudelijk begrijpen van teksten zijn hard nodig. Ondanks deze gebreken blijken auteurs en typisten positief te reageren op het nieuwe schrijfgereedschap. Kennelijk staan de beperkingen van de hedendaagse taaltechnologie toepassingen in de sfeer van intelligente tekstverwerking niet in de weg. Laat ik enkele voorbeelden geven. Het eerste betreft tekstwijzigingen op woordniveau. Stel, U wilt een zelfstandig naamwoord dat vaak in Uw tekst voorkomt, in het meervoud zetten. Een moderne tekstverwerker stelt U hiertoe in staat via één enkel commando. Maar de taalkundige veranderingen die deze wijziging oplegt aan andere zinsdelen, kunnen niet automatisch doorberekend worden. Bijvoorbeeld, *dit document* moet *deze documenten* worden. Bovendien moet in alle hoofd- en bijzinnen waar deze woordgroep de rol van onderwerp vervult, de persoonsvorm in het meervoud worden gezet (congruentieregel). Voorts moeten alle voornaamwoordelijke verwijzingen naar deze woordgroep (*het, dit, dat*) worden vervangen door een meervoudsvorm (*ze, deze, die,*). Op huidige tekstverwerkers moet dit met de hand gebeuren. In talen als Frans en Duits, die een rijk stelsel van woordvormingsregels (verbuigingen en vervoegingen) kennen, is dit een tijdrovende bezig-

INFORMATIE JAARGANG 29 NR. 5
PAG. 397 T/M 492 1987

heid die tot veel fouten aanleiding geeft. Toekomstige auteursystemen zullen dergelijke tekstwijzigingen automatisch kunnen uitrekenen.
Op zinsniveau wenst de auteur handige middelen om de volgorde van woorden en zinsdelen te veranderen in overeenstemming met taalregels (neem bijvoorbeeld het verschil in woordvolgorde tussen hoofd- en bijzinnen dat in het Nederlands en het Duits bestaat), om actief/passief-transformaties uit te voeren, om nevengeschikte zinsdelen in te voegen, om suggesties voor mogelijke parafrasen te opperen, enzovoort. Zulke faciliteiten vergen een geavanceerd gebruikersinterface dat op eenvoudige wijze toegang verschaft tot de taalkundige structuur van zinnen. Bijvoorbeeld moeten schrijvers de mogelijkheid hebben om 'syntactische bomen' op een grafisch beeldscherm zichtbaar te maken en om te manipuleren niet alleen met letters, woorden, zinnen en regels (zoals in hedendaagse tekstverwerkers), maar ook met zinsdelen als naamwoordsgroepen, voorzetselgroepen, bijzinnen en dergelijke. Auteursomgevingen zullen met name op het tekstniveau hun nut bewijzen. Een betrekkelijk eenvoudige toepassing is het automatisch voortbrengen van grote aantallen geïndividualiseerde documenten (zakenbrieven bijvoorbeeld) met een grotere verscheidenheid in zinsbouw en woordkeus dan mogelijk is in hedendaagse rapportgeneratoren en correspondentiesystemen die werken met min of meer vaste 'bouwstenen'.

4 Auteur- en dialoogsystemen tezamen

Aan de K.U. te Nijmegen bouwen we een prototype voor een gecombineerd auteur- en dialoogsysteem dat gebruik maakt van dezelfde linguïstische modules. Via het dialoogsysteem heeft de functionaris toegang tot relevante domeinkennis, bijvoorbeeld hoe bepaalde procedures in zijn kantoor dienen te verlopen. Een speciaal domein betreft de status van documenten die op een tekstverwerker worden aangemaakt, geredigeerd, (semi-)automatisch vertaald, nagekeken; en via elektronische post worden verstuurd naar collega's, door hen becommentarieerd, gefiatteerd enzovoort. Een typische vraag zou kunnen luiden: 'Heb ik de brief aan Mevr. X al afgemaakt en verstuurd?'. Het dialoogsysteem zoekt het betreffende document op en overhandigt het aan het auteursysteem. Dit zorgt er dan voor dat de tekst op het scherm wordt afgebeeld in een editor-raam.

Het ontwerp van ons systeem gaat ervan uit dat tekst op twee manieren kan worden behandeld: enerzijds als een reeks van karakters – opgeslagen in een bestand, afgebeeld op het scherm –, anderzijds als een linguïstisch gestructureerd object. Beide representaties van de tekst worden in onderlinge samenhang bewaard en gemanipuleerd. Stel bijvoorbeeld dat de schrijver het auteursysteem opdraagt een passieve zin actief te maken. (Hij doet dit door een commando te geven aan de 'boom-editor' die tot taak heeft taalkundige structuren, in het bijzonder syntactische bomen, te behandelen.) Het auteursysteem reageert dan door wijziging van:
1 de linguïstische representatie van de zin;
2 de corresponderende karakterreeks in het tekstbestand; en
3 de afbeelding op het scherm.

Aan de andere kant, wanneer de schrijver de afbeelding van de zin op het beeldscherm rechtstreeks wijzigt, bijvoorbeeld door aan een zelfstandig naamwoord een meervoudsuitgang toe te voegen, zal het auteursysteem ertoe overgaan de corresponderende verandering in de taalkundige representatie aan te brengen alsmede alle verdere veranderingen die de grammatica voorschrijft. De bijbehorende orthografische weergave van de zin wordt tenslotte in het editor-raam afgebeeld.
Aldus kunnen schrijvers hun tekst op twee manieren bijstellen. Ze kunnen de *orthografische* representatie wijzigen door rechtstreeks in een tekstbestand te typen – de methode die in huidige tekstverwerkers gevolgd wordt. Maar bovendien kunnen ze – via de boomeditor – wijzigingen aanbrengen in de *linguïstische* representatie van de tekst. In beide gevallen ziet het auteursysteem erop toe dat niet alleen de door de schrijver zelf aangebrachte wijzigingen worden opgenomen in het tekstbestand maar ook de verdere wijzigingen die taalkundig zijn geïmpliceerd.
Ik vind dat tekstverwerkingsfaciliteiten van het hier beschreven type thuishoren in elk kantoorwerkstation dat het epitheton 'vijfde generatie' verdient. Ook ben ik van mening dat de voorgestelde tekstverwerkingstechniek die gebaseerd is op twee onderling gekoppelde tekstrepresentaties – één orthografisch, één linguïstisch – essentieel afwijkt van zelfs de meest geavanceerde hedendaagse tekstverwerkers. Dit is een extra reden om aan 'auteursysteem' het epitheton 'vijfde generatie' mee te geven: als de opvolger namelijk van vier eerdere generaties, te weten handschrift, boekdrukkunst, schrijfmachine, en tekstverwerking in haar huidige gedaante.

5 Postcriptum: Een intelligente studie-tekstverwerker

Van vele zijden komen berichten over de veel te wensen overlatende schrijfvaardigheid van Nederlanders. Het gaat hierbij niet alleen om leerlingen van het basisonderwijs maar ook om scholieren, studenten en afgestudeerden. Voorstellen voor oplossing van dit probleem zijn veel minder talrijk. Opvoering van het aantal schriftelijke werkstukken dat tijdens een opleiding gemaakt moet worden, zou momenteel een aanzienlijke taakverzwaring inhouden voor student zowel als docent.
Nu studenten steeds gemakkelijker toegang krijgen tot rekenapparatuur, zien we dat schriftelijke werkstukken (essays, practicumverslagen, brieven, scripties, tentamens) steeds vaker vervaardigd worden met behulp van een tekstverwerker. Waarschijnlijk zal dit op zichzelf al leiden tot verbetering van het schrijfproces: nettere opmaak van het eindprodukt, meer efficiënte behandeling van voorlopige versies. Niettemin kunnen huidige tekstverwerkers nauwelijks ondersteuning verlenen bij het plannen van de opbouw van een tekst, en bij het bewaken van de taalkundige kwaliteit.

INFORMATIE JAARGANG 29 NR. 5
PAG. 397 T/M 492 1987

Belangrijke ontwikkelingen op deze twee gebieden zijn 'outliners' als ThinkTank en Framework die het concipiëren van inhoud en structuur aanzienlijk vergemakkelijken, en auteursomgevingen voor spelling-, grammatica- en stijlbewaking. Beide typen van systemen openen interessante perspectieven voor onderwijstoepassingen. Maar realisatie daarvan ten behoeve van universitaire en andere opleidingen vraagt eerst nog aanzienlijke aanpassingen en aanvulling met nieuwe principes.

Men zou een intelligente studie-tekstverwerker kunnen bouwen die bruikbaar is voor schriftelijk werk dat deel uitmaakt van individueel onderwijs (scripties bijvoorbeeld) of cursorisch onderwijs (zoals practicumverslagen). Met het oog op het laatste zal het gewenst zijn dat de tekstverwerker opgenomen is in een netwerk waarop ook de docent (en de cursusadministratie) is aangesloten. De studie-tekstverwerker zal onder meer de volgende functionaliteit moeten bieden:
– in beeld brengen van de structuur van een tekst met behulp van bomen en/of graphen (netwerken). Onder structuur versta ik niet alleen de hoofdstuk- of paragraafindeling, maar ook de inhoudelijke opbouw (een gevolgde argumentatie bijvoorbeeld). Grafische programmatuur moet de student in staat stellen de structuur expliciet aan te brengen in de vorm van een boom of een graph (met gelabelde knopen) die op het scherm getekend kan worden en daarna gemodificeerd. Deze grafische weergave moet gekoppeld zijn aan de interne orthografische tekstrepresentatie, ongeveer op de manier van bestaande outliners. De docent kan een door hem/haar gewenste tekststructuur vooraf grafisch uitwerken en als een 'model' aan de student aanbieden. (Een grafische boom-editor die deze toepassing in beginsel mogelijk maakt, is ontwikkeld door Desain, 1985.)
– grammatica-, spelling en stijlbewaking voor het Nederlands, dat wil zeggen een Nederlandstalige auteursomgeving.

Dergelijke voorzieningen zullen het docenten aanzienlijk makkelijker maken schriftelijke werkstukken op te geven, na te kijken en te verbeteren. Studenten zullen ze in kortere tijd en beter gemotiveerd kunnen schrijven. De bewakingsfuncties zullen de student attenderen op taalkundige fouten. De docent krijgt niet alleen een 'platte tekst' onder ogen maar kan bijvoorbeeld ook de *tekststructuur* zichtbaar maken die door de student is bedacht. Aldus is een snelle globale beoordeling mogelijk. Ook kan de docent gemakkelijk commentaar invoegen bij afzonderlijke passages (knopen) zonder de tekstopmaak van het werkstuk te beïnvloeden.

Verdere uitbreiding van deze functionaliteit is denkbaar. Bijvoorbeeld toegang vanuit de studie-tekstverwerker tot grote literatuur- en databestanden die de student moet raadplegen. En op de langere duur: een (semi-)automatische beoordeling van de inhoudelijke structuur van een werkstuk op basis van de gelabelde boom/graph. Hierbij kunnen Artificiële-Intelligentietechnieken wellicht uitkomst bieden.

LITERATUUR

Vertaling van het artikel *Beyond Word Processing* dat is verschenen in: G. Bunting (Ed.), Information Management Yearbook 1986 (London: Institute of Data Processing Management). Deze tekst is samen met andere geschriften van de auteur samengebracht in de bundel *Natuurlijke Taal en Kunstmatige Intelligentie: Taal tussen mens en machine* die in april 1987 bij Wolters-Noordhoff te Groningen is verschenen. De in dit artikel verwoorde ideeën danken hun ontstaan deels aan discussies met Gert Anbeek, Walter Daelemans, Peter Desain, Leo Konst, Fieny Pijls, Henk Schotel en Koenraad De Smedt.

Desain, P., *TREE DOCTOR: a software package for graphical manipulation and animation of tree structures*. Voordracht voor het Symposium over Mens-Machine Interactie, Amsterdam, December 1985. (Gaat verschijnen in de Proceedings. Ook beschikbaar als intern rapport van het Psychologisch Laboratium KU Nijmegen, no. KUN-86-CW01.)

Frase, L.T., 'The UNIX Writer's Workbench Software: Philosophy'. *The Bell System Technical Journal*, 1983, 62, 1883-1890.

Heidorn, G.E., K. Jensen, L.A. Miller, R.J. Byrd, & M.S. Chodorow, 'The EPISTLE text-critiquing system'. *IBM System Journal*, 1982, 21, 305-326.

Hendrix G., 'Natural-Language Interface'. *American Journal of Computational Linguistics*, 1982, 8, 56-61.

Wilensky, R., *Talking to UNIX in English: an overview of an on-line consultant*. Report No. 82/104, Computer Science Division, University of California, Berkeley, september 1982.

Gerard Kempen is als hoogleraar taalpsychologie verbonden aan Psychologisch Laboratorium van de KU Nijmegen.

Tabel 1: Karakteristieke functionaliteit van niveau-1-dialoogsystemen (vrij naar Hendrix, 1982)

1	Feitenvragen beantwoorden die betrekking hebben op gegevens in de gebruikelijke typen databanksystemen.
2	Hierbij desgewenst datafiles 'coördineren' (bijvoorbeeld 'What is Smith's location?' wordt vertaald in ‚What is the location of the department of Smith?').
3	Eenvoudige gevallen van pronominalisatie oplossen (dat wil zeggen achterhalen waar voornaamwoorden als *het, hij, dat, haar* enzovoort naar terugverwijzen; moeilijke gevallen blijven echter buiten bereik).
4	Eenvoudige gevallen van ellipsis behandelen (elliptische vragen zijn onvolledig en grijpen terug op voorafgaande zinnen in de dialoog: 'Waar is Jan?'... 'En Piet?').
5	Coöperatieve antwoorden geven op 'nul-vragen' (bijvoorbeeld op de vraag 'Hoeveel exemplaren van boek X heb je in voorraad?' antwoordt het systeem niet met 'Geen enkel' maar met 'Ik ken dat boek helemaal niet' of 'Dat boek is uitverkocht').
6	De taalkennis verrijken in interactie met de gebruiker (bijv: Define 'JD' as 'Jefferson Davis Jones'. Let 'Q1 Smith salary' be like 'What is the salary of employee Smith?' 'Q1 JD AGE?').
7	Invoerzinnen (vragen, opdrachten e.d.) parafraseren (zodat de gebruiker enigszins kan controleren of ze goed zijn overgekomen).
8	Spelfouten corrigeren en zinvol reageren op ongrammaticale invoerzinnen.
9	Gegevens bijwerken ('updating') d.m.v. natuurlijke taal (bijv. 'Change Bob Day's location to Building 7.').

10 Beantwoording van 'meta-vragen' (bijvoorbeeld 'What information is in the database?', 'What are the allowable values for employee job titles?', 'How timely are the sales data?', 'Can you handle relative clauses?').

Tabel 2: Voorbeeld van een dialoog met Wilensky's (1982) Unix Consultant

– *How do I delete a file?*
Typing 'rm filename' will remove the file with name filename from your directory.

– *A directory?*
Typing 'rm -r directoryname' will recursively delete the directory with name directoryname and all the files and directories contained in the directory.

– *How should one edit a file?*
Typing 'ex filename' will get you into the ex editor; typing 'emacs filename' will get you into the emacs editor.

– *What is the best way to create a file?*
The best way to create a file is to use the editor, i.e. 'ex filename'.

– *A directory?*
Typing .mkdir directoryname' will create a directory with name directoryname in your current directory.

– *How can I find out how full the disk is?*
Typing 'df' will tell you how full the disk is.

– *What is a search path?*
A search path in Unix is a list of directories in which the operating system searches for programs to execute.

INFORMATIE JAARGANG 29 NR. 5
PAG. 397 T/M 492 1987