

Auteursomgevingen: Tekstverwerkers van de vijfde generatie

Gerard Kempen, Gert Anbeek, Peter Desain, Leo Konst, en Koenraad de Smedt

Technieken die binnen de **Artificiële Intelligentie (AI)** ontwikkeld zijn voor verwerking van natuurlijke taal, maken de bouw mogelijk van 'intelligente' tekstverwerkers die nieuwe vormen van ondersteuning bieden bij het schrijven, redigeren en vertalen van teksten en documenten. Als algemene benaming voor zulke tekstverwerkers stellen wij de termen *auteursomgeving* of *auteursysteem* voor.

Na een theoretische uiteenzetting over het representeren van in natuurlijke taal gestelde tekst in een computer, beschrijven wij achtereenvolgens doelstelling, ontwerp, implementatie en functionaliteit van de auteursomgeving die in het Psychologisch Laboratorium van de KUN wordt ontwikkeld als onderdeel van een ESPRIT-project dat de bouw van een intelligent kantoorwerkstation beoogt. De aandacht zal met name gericht zijn op de linguïstische modules en het gebruikersinterface.

1 Inleiding

AI-technieken voor de verwerking van natuurlijke taal kunnen de bouw mogelijk maken van meertalige, 'intelligente' tekstverwerkers die nieuwe vormen van ondersteuning bieden bij het schrijven, redigeren en vertalen van teksten en documenten. Als algemene benaming voor zulke tekstverwerkers stellen we de termen *auteursomgeving* of *auteursysteem* voor. Hieronder verstaan wij alle programmatuur die hulp biedt aan hen die in een kantooromgeving belast zijn met het schrijven en redigeren van in natuurlijke taal gestelde teksten (zie ook Kempen, 1987).

De grootste tekortkoming van bestaande tekstverwerkers en tekst-editors is dat zij vrijwel verstoten zijn van kennis op syntactisch gebied. Als zij al enige taalkundige kennis bezitten, dan blijft deze beperkt tot spelling en woordafbreking. Er komt evenwel verbetering in deze situatie, nu diverse programmapakketten voor tekstbewaking, waaronder de *Writer's Workbench* van AT&T (Fraser, 1983) en *CRITIQUE* van IBM (Heidorn, Jensen, Miller, Bird & Chodorow, 1982) op de markt zijn gebracht. Deze pakketten kunnen gebruikt worden voor de verbetering van spelling, grammatica en stijl van in het Engels geschreven teksten. De geboden mogelijkheden variëren van controle op woordafbreking, interpunctie en spelling tot controle op grammatica en stijl ('split infinitives', te veel passief, incongruentie; breedsprakigheid, zinslengte, abstractieniveau). Een andere categorie hulpmiddelen wordt gevormd door de zogenaamde '*idea processors*' of '*outliners*'. Zij vereenvoudigen de inhoudelijke en retorische aspecten van het schrijfproces. Een karakteristiek voorbeeld hiervan is *ThinkTank* (zie bijvoorbeeld Kamin, 1986), een programma dat schrijvers helpt om al in het beginstadium een opzet voor de gehele tekst uit te werken. Deze tekstopzet kan op het

scherm worden afgebeeld als een boomstructuur, waarvan de knopen corresponderen met hoofdstukken, paragrafen, alinea's, enz.

In dit artikel wordt het ontwerp en de gedeeltelijke implementatie beschreven van een auteursomgeving, waarin de faciliteiten van de huidige tekstverwerkers (bewerking van tekstfiles, tekstopmaak) volledig geïntegreerd zijn met de faciliteiten op linguïstisch en retorisch niveau. Na een theoretische uiteenzetting over het representeren van in natuurlijke taal gestelde tekst in een computer, volgt een schets van het auteursysteem. Doelstelling, ontwerp, implementatie en functionaliteit hiervan worden beschreven in paragrafen 3 tot en met 6.

2 Tekstrepresentatie

Wij kunnen een in natuurlijke taal gestelde tekst vanuit tenminste vier verschillende gezichtspunten beschouwen:

- als een reeks (letter)tekens ('grafemen');
- als een document met een bepaalde uiterlijke vormgeving;
- als een verzameling linguïstische eenheden (woorden, zinsdelen, zinnen), en
- als een conceptuele/retorische structuur.

Deze gezichtspunten duiden wij verder aan met de respectievelijke termen *grafemisch*, *typografisch*, *linguïstisch* en *retorisch*. Moderne tekstverwerkers zijn nog absoluut niet in staat om volledige ondersteuning te bieden op alle vier deze gebieden tegelijk. Zij overschrijden nauwelijks het grafemische en typografische niveau. Outline-generatoren combineren het grafemische met het retorische aspect. Een beperkt linguïstisch perspectief op de tekst wordt geboden door tekstverwerkers die spelling en zinsbouw controleren. Auteursomgevingen zoals door ons beoogd, moeten een zeer rijke, veelzijdige representatie kunnen opbouwen van teksten in natuurlijke taal. In deze representatie dienen alle relevante aspecten die vanuit de vier gezichtspunten verschijnen op systematische wijze tot uitdrukking te komen. Tabel 1 geeft een lijst van aspecten, die wij kort zullen typeren.

Vanuit *retorisch* perspectief worden twee mogelijke typen van structuren zichtbaar: de inhoudelijke (bijvoorbeeld de opeenvolging van gebeurtenissen, de opbouw van een argument, oorzaak-gevolgrelaties), en de formele, waarbij wij denken aan hoofdstukken, paragrafen, alinea's, enz. Omdat deze laatste structuur strikt hiërarchisch van aard is, kan zij als een boom-

<i>Gezichtspunt</i>	<i>Aspect</i>
Retorisch	Organisatie van de inhoud Organisatie in paragrafen ('tekst-schema')
Linguïstisch	Semantiek (betekenis) Syntaxis (zinsbouw) Lexicon (woordenschat) Morfologie (woordopbouw) Fonologie (uitspraak)
Typografisch	Vormgeving van het document Bladspiegel Lettertypen
Grafemisch	Tekstfiles Karakterreeksen

Tabel 1: Tekstrepresentatie: enkele gezichtspunten en aspecten.

structuur worden afgebeeld. De inhoudelijke structuur vraagt echter om een veel krachtiger representatieschema. Cyclische grafen (netwerken) lijken hier te verkiezen boven boomstructuren (zie Brown, 1986). Vanuit *linguïstisch* perspectief onderscheiden wij de vijf welbekende structuurlagen. (Verderop zal duidelijk worden welke taken wij deze lagen toewijzen; zie ook Kempen, Konst & De Smedt, 1984.) Het *typografische* gezichtspunt heeft vooral betrekking op de uiterlijke vorm van het document als geheel (bijvoorbeeld boek, tijdschrift of krant), de bladspiegel (met inbegrip van figuren en tabellen) en lettertypen (font, grootte, stijl). Vanuit *grafemisch* perspectief is de tekst primair een file met karakterreeksen, welke veranderd en doorzocht kunnen worden (zoals in traditionele tekst-editors).

Ieder gezichtspunt moet op twee manieren gerepresenteerd worden: de interne, *symbolische* representatie die gemanipuleerd wordt door de tekstverwerkingssoftware, en de externe, *visuele* representatie die grafisch wordt afgebeeld en die door de gebruiker kan worden bekeken. Zo kan bijvoorbeeld de syntactische structuur van een zin, die intern als een lijst gerepresenteerd wordt, aan de gebruiker getoond worden in de vorm van een boomdiagram. En de structuur van een argument dat door de schrijver wordt uitgewerkt, kan als een netwerk op het scherm worden afgebeeld. Een tekstrepresentatie met de hier voorgestelde complexiteit kan computationeel gerealiseerd worden als een grote verzameling objecten van zeer verschillende aard (morfemen, woorden, regels, paragrafen, alinea's, knopen, verbindingen, boomstructuren, enz.). Ieder type object wordt gedefinieerd als een lijst van paren die uit een kenmerk en een waarde bestaan, waarbij sommige waarden verwijzen naar andere objecten. Zo ontstaat een ingewikkeld netwerk van onderling gerelateerde knopen, dat behandeld wordt door object-georiënteerde programmatuur, zoals een syntactische ontleder en generator, programma's voor tekstopmaak en programma's voor de bewerking van grafische boomstructuren en netwerken.

3 Doelstellingen

Wij ontwikkelen een Auteursomgeving die aan de volgende eisen moet voldoen:

1. Het systeem moet in staat zijn de in het vorige hoofdstuk beschreven veelzijdige tekstrepresentaties op te bouwen en er bewerkingen op uit te voeren. Wijzigingen in de tekst die vanuit welk perspectief dan ook worden aangebracht, moeten automatisch worden doorberekend naar de andere gezichtspunten, zodat een consistent geheel behouden blijft.
2. Het Auteursysteem moet de schrijvers in staat stellen de tekst vanuit ieder afzonderlijk gezichtspunt te bekijken, en iedere gewenste combinatie van gezichtspunten en aspecten samen te stellen die logisch mogelijk is. Binnen een gekozen gezichtspunt staan de auteur doeltreffende middelen (menu's, commando's, directe manipulatie van objecten) ter beschikking om bewerkingen op de tekst uit te voeren.
3. Het gebruikersinterface moet transparant zijn in die zin dat de gebruiker zich niet bewust hoeft te zijn van de veelzijdige interne tekstrepresentatie en de corresponderende gezichtspunten.
4. De linguïstische en typografische componenten moeten aangepast kunnen worden aan andere talen en schriftsoorten dan de huidige (Nederlands).
5. Het systeem moet niet alleen in staat zijn teksten te bewerken die door auteurs worden geschreven, maar het moet ook op basis van semantische specificaties automatisch (semi-standaard) teksten kunnen genereren.
6. Het systeem zal alleen oppervlakkige betekenisrepresentaties bevatten, zodat geen grote hoeveelheden 'kennis van de wereld' hoeven te worden ingevoerd. Dit impliceert dat faciliteiten ontbreken die in hoge mate afhankelijk zijn van diepe semantische representaties (bijvoorbeeld het maken van samenvattingen).

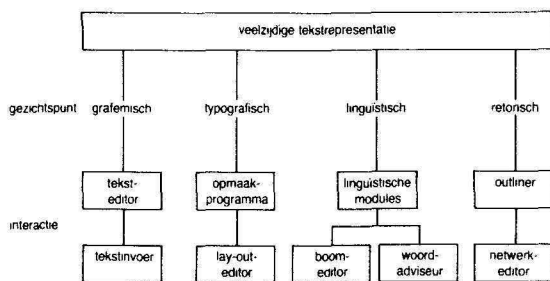
4 Ontwerp

Figuur 1 geeft het basisontwerp van het Auteursysteem weer. Met ieder gezichtspunt dat wij onderscheiden, correspondeert een afzonderlijke programma-component die toegang heeft tot en bewerkingen kan uitvoeren op bepaalde objecten in de tekstrepresentatie. De diverse componenten interacteren op velerlei wijze. Zo kan de auteur bijvoorbeeld een opzet van een nieuwe tekst maken (tekstschema) door de Netwerkditor aan te roepen. Deze voorziet hem van een 'kladblok', waarop hij een netwerk van knopen en verbindingen kan tekenen, welke overeenkomen met enkelvoudige concepten en hun onderlinge relaties. Hiervoor hoeft hij slechts een paar eenvoudige handelingen met de muis uit te voeren. Op deze manier specificeert de auteur een eerste retorische structuur van de tekst. Door op één van de knopen te 'klikken' kan hij daarna de Tekstverwerker aanroepen (grafemisch gezichtspunt), die dicht bij de betreffende knoop een venster (window) laat verschijnen. In dit

venster kan de eerste versie getypt worden van de tekst die betrekking heeft op het geselecteerde deel van het tekstschema. Binnen het tekstvenster kan het linguïstisch gezichtspunt worden gekozen door bijvoorbeeld de spellingchecker aan te roepen voor het zichtbare deel van de tekst. Hiertoe moet de Woordadviseur geactiveerd worden. Een andere mogelijkheid is de Zinsontleder aan te roepen (zie hierna), waarmee de syntactische structuur van een deel van de tekst geïnspecteerd kan worden. De Boomeditor beeldt één of meer syntactische bomen op het scherm af. In het geval van een onjuiste ontleding (zie figuur 2) kan de auteur deze grafisch wijzigen. Hij kan ook de zin veranderen door bijvoorbeeld (met behulp van de muis) een deelboom op te pakken en hem op een andere plaats in de zin aan te hechten. De nieuw gevormde zin kan door de Zinsontleder gecontroleerd worden. Als deze geen onrechtmatigheden ontdekt, wordt in de tekst de oorspronkelijke zin door de nieuwe vervangen.

Nadat hij is teruggekeerd in het tekstvenster (grafisch gezichtspunt), kan de auteur het typografische gezichtspunt kiezen door de Lay-out-editor aan te roepen, waarin hij de gewenste bladspiegel, het gewenste lettertype, enz. kan specificeren. Als de auteur de Tekstverwerker verlaat (het tekstvenster verdwijnt), keert hij terug naar het tekstschema (retorisch gezichtspunt). Door op een andere knoop in het netwerk te klikken, opent de auteur een leeg tekstvenster, waar hij de bij die knoop behorende tekst kan intikken. Een alternatieve mogelijkheid is dat hij in de Netwerkeditor blijft om het tot dan toe ontwikkelde tekstschema te reorganiseren door knopen en verbindingen toe te voegen. Wil hij de definitieve indeling van de tekst in paragrafen vastleggen, dan moet de auteur het schema onderbrengen in een hiërarchische boomstructuur (inclusief de links-rechtsvolgorde van de takken). Op grond van deze structuur berekent het opmaakprogramma de juiste volgorde en opmaak van de verschillende tekstdelen.

Wij hopen dat dit voorbeeld de interpretatie van het basisontwerp zoals wij dat in Figuur 1 hebben geschetst, vereenvoudigt. Wij moeten echter nog vermelden dat in het Auteursysteem dat wij bezig zijn te ontwikkelen (zie volgende paragraaf) niet alle verbindingen tussen de componenten geïmplementeerd zullen worden. Het ontwerp is vooral bedoeld als conceptueel hulpmiddel om in de toekomst beter gestructureerde, meer modulair opgebouwde tekstverwerkers te ontwikkelen.



Figuur 1: Basisontwerp van de Auteursomgeving

5 Implementatie

Bij de ontwikkeling van de Auteursomgeving hebben wij ons met name gericht op de linguïstische aspecten, op het gebruikersinterface (vooral op de grafische editors) en op de integratie van beide. Wij hebben Nederlands gekozen als doeltaal, maar tevens rekening gehouden met de wenselijkheid het systeem naar andere talen uit te kunnen breiden. Het systeem draait op een Symbolics Lisp Machine en het maakt gebruik van de daarop aanwezige faciliteiten voor tekstverwerking (Emacs), schermindeling (windowing) en typografie. Alle programma's zijn geschreven in Common Lisp, onder gebruikmaking van object-georiënteerde uitbreidingen daarvan: FLAVORS en ORBIT (De Smedt, 1987).

In deze paragraaf geven wij een korte beschrijving van de verschillende linguïstische modules en van het gebruikersinterface.

Editors voor grafische netwerken en boomstructuren

Desain heeft een softwarepakket ontwikkeld voor de grafische bewerking, manipulatie en animatie van netwerken (grafan). Dit geeft de gebruiker de mogelijkheid om te gaan met knopen en verbindingen als waren het echte voorwerpen ('directe manipulatie'; zie Shneiderman, 1982). Dit pakket bevat een boomeditor en een routine voor het tekenen van boomstructuren (Desain, 1987).

Woordenboek

Wij hebben een on-line woordenboek opgezet (werk van H. Schotel en E.-J. van der Linden) waarin woorden zeer snel worden gevonden. Het bevat alle woordkenmerken die de andere linguïstische modules nodig hebben. De meeste verbuigingen en vervoegingen (meer dan 50 000) van 10 000 gangbare Nederlandse woorden staan erin (bijvoorbeeld het enkel- en meervoud van zelfstandige naamwoorden en de vlekkenvormen hiervan). Samen met de citatievorm waarvan zij zijn afgeleid, zijn deze vormen opgeslagen in een indexed-sequential file. Iedere woordvorm (bijv. *liep*) is gerepresenteerd als een record dat verwijst naar een citatievorm (bijv. *lopen*). Dit garandeert dat de kenmerken (woordklasse, subcategorisatie, morfologische opbouw, klemtoon) van een eenmaal gevonden woord in enkele tientallen milliseconden beschikbaar komen. De records zijn toegankelijk via twee aparte indexen. De eerste maakt het mogelijk zeer snel inflecties van een gegeven citatievorm op te zoeken; de tweede geeft de citatievorm die bij een gegeven woordvorm hoort. Op deze wijze kan het woordenboek gebruikt worden bij zowel het genereren als het analyseren van taal.

Woordafbreking

Wij hebben samen met W. Daelemans een algoritme ontwikkeld en geïmplementeerd voor woordafbreking. Het algoritme kan de gehele potentiële woordenschat van het Nederlands vrijwel foutloos opsplitsen in lettergrepen: niet alleen bestaande woorden, maar ook nieuwe samenstellingen. Op deze woorden past het een gedeeltelijke morfologische ontleding toe (opsplitsen van samenstellingen, afsplitsen van affixen; zie het

voorbeeld hierna). Deze is noodzakelijk omdat morfologische grenzen prioriteit hebben boven fonologische regels voor lettergreepsplitsing. Deze laatste regels zouden bijvoorbeeld het zelfstandig naamwoord *inventionsoproep* opsplitsen in *i-nen-ting-sop-roep*. De morfologische ontleder echter herkent *inenting* en *op-roep* als zelfstandige naamwoorden, *in* en *op* als voorvoegsels, en *s* als een bindmorfeem. De juiste opsplitsing is derhalve *in-en-tings-op-roep*.

Woordherkenning (spellingcontrole en spelfoutcorrectie)

Ons programma voor spellingcontrole voor het Nederlands is geschreven door W. Daelemans. Allereerst gaat het na of de ingevoerde letterreeks voorkomt in voornoemd woordenboek of in een speciaal door de gebruiker samengesteld woordenboek. Als dit niet het geval is, wordt een morfologische analyse geprobeerd. Dit is noodzakelijk omdat in het Nederlands (evenals in het Duits, maar niet in het Frans en Engels) bestaande woorden bijna onbeperkt tot nieuwe woorden kunnen worden samengevoegd. Zonder morfologische analyse zou iedere nieuwe samenstelling als fout aangemerkt worden. (In feite gebeurt dit bij vrijwel alle bestaande commerciële spellingcheckers voor het Nederlands). Als ook de morfologische analyse faalt, verschijnt een foutmelding op het scherm.

Omdat van het Auteursysteem méér verwacht mag worden dan alleen spelfouterkenning, heeft Van Berkel (1986) een component ontwikkeld voor de *correctie* van spelfouten. Deze module is in staat juiste spellingen te suggereren. Eén deelcomponent hiervan is een algoritme voor (gedeeltelijke) *grafeem-foneem-omzetting*. Dit beeldt al dan niet correct gespelde woorden af op foneemreeksen die de uitspraak (of uitspraakmogelijkheden) van die woorden op een gestandaardiseerde wijze weergeven. Het systeem zoekt dan de bijbehorende juiste spelling op. Zo wordt de achternaam *Craeybeckx* herkend in de onjuiste maar homofone spelling *Kraaibeks*.

Deze correctiecomponent hebben wij op twee manieren uitgebreid. De ene heeft tot doel de criteria voor herkenning soepeler te maken via berekening van gelijkheidsmaten tussen foneemreeksen (de zg. 'trifonemmethode'). In de andere wordt rekening gehouden met de Nederlandse spellingregels die afhankelijk zijn van de syntaxis (m.n. betreffende werkwoordsvormen).

Zinsontleding en zinsgeneratie

De ontleder, geschreven door Konst, gebruikt een all-path ontleedalgoritme, dat een bottom-up, breadth-first, left-to-right strategie volgt. Aan de paden worden stochastische waarden toegekend, die gespecificeerd zijn in de herschrijffregels van de grammatica. Deze waarden worden gebruikt om aan iedere ontleding een waarschijnlijkheidswaarde toe te kennen. Deze waarschijnlijkheidswaarde speelt een belangrijke rol bij de behandeling van ambiguïteit en bij de manier waarop met ongrammaticale zinnen wordt omgesprongen (robuustheid, flexibiliteit). De syntactische structuur die voor ingevoerde zinnen berekend wordt,

maakt gebruik van dezelfde notatie als de zinsgenerator (Incrementeel-Procedurele Grammatica; Kempen & Hoenkamp, 1987). De ontleder kan in korte tijd (in de orde van een seconde per woord) tamelijk ingewikkelde structuren verwerken. De door de ontleder berekende syntactische bomen worden omgezet in expressies uit een semantische-representatietaal die wij verderop beschrijven.

De door De Smedt (in voorbereiding) geschreven zinsgenerator kan niet alleen volledige zinnen produceren op basis van ingevoerde semantische representaties, maar hij kan ook globale en lokale veranderingen aanbrengen in bestaande zinnen. De noodzaak voor dit laatste ontstaat bijvoorbeeld als de gebruiker hierom expliciet verzoekt (bijv. 'Zet deze zin in de bedrijvende vorm'), of als er een verandering optreedt in de onderliggende semantische representatie van een bestaande zin. De generator probeert zulke veranderingen aan te brengen door de bestaande zin te wijzigen in plaats van hem van de grond af aan opnieuw op te bouwen. (Dit laatste zou een parafraze met een totaal andere structuur als ongewenst gevolg kunnen hebben).

Semantische Representatie

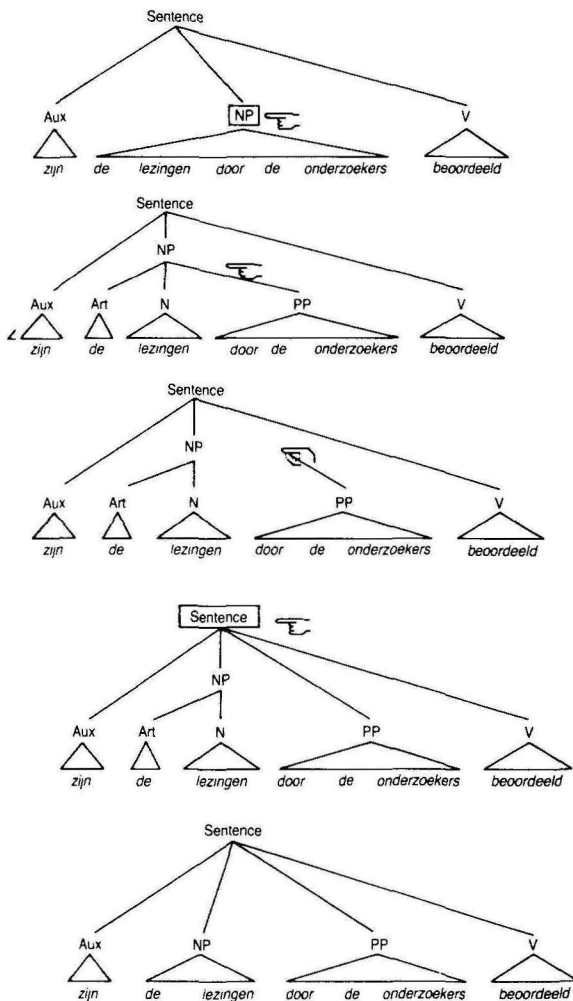
Wij hebben een semantische-representatietaal ontworpen (Geurts, 1986) die de rol kan vervullen van intermediair tussen de ontleder en de generator. Zowel de uitvoer van de ontleder als de invoer van de generator bestaat uit uitdrukkingen in deze taal. De taal maakt gebruik van een tamelijk algemene *case-frame-notatie*. De betekenisrepresentaties zijn *oppervlakkig*: er wordt bijvoorbeeld een één-op-één relatie voorondersteld tussen inhoudswoorden (van het Nederlands) en concepten (van de representatietaal). Coreferentie-relaties (bijvoorbeeld tussen voornaamwoorden en hun antecedenten) worden echter expliciet aangegeven. Op dit moment heeft nog geen enkele component zijn definitieve vorm bereikt. Voorts hebben wij bij bovenstaande opsomming niet gestreefd naar volledigheid. On-line vertaalwoordenboeken hebben wij bijvoorbeeld buiten beschouwing gelaten.*

6 Functionaliteit

Het zou voorbarig zijn om nu al gedetailleerde specificaties te geven van alle diensten en functies die voor de gebruiker beschikbaar zullen komen. Gezien het grote aantal mogelijkheden is zowel een zorgvuldig ontwerp nodig, als een uitgebreide toetsing met verschillende soorten gebruikers (zoals typisten, schrijvers, professionele redacteurs, vertalers) met verschillende opleidingsniveaus en uiteenlopende grammaticale kennis. In plaats van ideeën weer te geven die in de literatuur geschetst (en gedeeltelijk in moderne tekstverwerkers geïmplementeerd) zijn, beschrijven wij hier enkele nieuwe mogelijkheden voor *tekstbewerking* en *-bewaking*.

De gebruiker die een regelmatig in de tekst voorkomend zelfstandig naamwoord in het meervoud wil zetten, kan dit op een hedendaagse tekstverwerker doen via één enkel commando. De linguïstische veranderingen die deze wijziging nodig maakt in andere zinsde-

len kunnen echter niet automatisch berekend worden. Zo moet een gaaf exemplaar veranderd worden in gave exemplaren. Ook moet in iedere zin waar deze naamwoordsgroep de functie van grammaticaal onderwerp vervult, de persoonsvorm in het meervoud gezet worden om de congruentie te handhaven. Bovendien moeten alle voornaamwoorden die verwijzen naar die naamwoordsgroep aangepast worden (*het, dit* en *zijn* moeten worden vervangen door *zij/hen, deze* en *hun*). In moderne tekstverwerkers moeten deze wijzigingen met de hand worden aangebracht – een tijdrovend proces dat bovendien veel fouten tot gevolg heeft in talen met een rijke morfologie zoals het Frans of Duits. In een Auteursomgeving worden deze veranderingen automatisch doorberekend.



Figuur 2: Directe manipulatie van een zinsstructuur met behulp van de Boomeditor. Vijf momentopnamen laten zien hoe een gebruiker een ontfeetboom van een ambigue zin bewerkt. In dit voorbeeld heeft de zinsontleder de ambiguïteit niet herkend, en de boomstructuur opgeleverd die hoort bij de interpretatie: 'Zijn de lezingen, gehouden door de onderzoekers, beoordeeld?' (boom 1). De gebruiker die bedoeld heeft: 'Zijn de lezingen beoordeeld door de onderzoekers?', kan de structuur gemakkelijk veranderen door enige handelingen met behulp van de muis uit te voeren. Het resultaat van de verandering is geschikt voor verdere bewerking (passief-actief transformatie; tekst-spraak omzetting; vertaling).

Het 'grammatica-spreadsheet' dat ons hier voor ogen staat zal af en toe fouten maken wanneer de ontleder een zin onjuist geanalyseerd heeft. In dergelijke gevallen kan de auteur met behulp van de Boomeditor de noodzakelijke verbeteringen in de boomstructuren aanbrengen (vergelijk Figuur 2). Dit extra werk loont de moeite indien;

- de auteur zich werkelijk zorgen maakt over de kwaliteit van de tekst (vooral wanneer hij in een vreemde taal schrijft);
- het document vele malen opnieuw gebruikt wordt met slechts kleine linguïstische veranderingen (bijvoorbeeld semi-standaard zakenbrieven), of
- het aantal lexicale en syntactische ambiguïteiten tot een minimum beperkt moet worden (bijvoorbeeld in contracten of wetsteksten).

Een ander voordeel van deze manier van tekstverwerking is dat auteurs de mogelijkheid hebben complexe operaties op hele zinnen uit te voeren door middel van eenvoudige opdrachten. Wij denken hierbij aan het samenstellen en splitsen van zinnen (neem bijvoorbeeld het verschil in verplichte woordvolgorde tussen hoofd- en bijzinnen in zowel het Nederlands als het Duits), aan het wijzigen van de lijdende vorm in de bedrijvende, of aan parafraseren.

Auteurs die hun teksten vervaardigen met behulp van een Auteursysteem hebben een belangrijk aanvullend voordeel als zij andere linguïstische bewerkingen van hun schrijfproducten voorzien: automatische vertaling of tekst-naar-spraak omzetting. Goede zinsontledingen zijn een eerste vereiste bij deze bewerkingen (met name bij vertaling, wat minder bij tekst-naar-spraak omzetting). Als de ontledingen al uitgevoerd en gecontroleerd zijn onder verantwoordelijkheid van de auteur, is een belangrijk obstakel op het pad naar succesvolle vertaling en tekst-naar-spraak omzetting uit de weg geruimd. Deze bewerkingen worden veel eenvoudiger wanneer betrouwbare informatie over de structurele eigenschappen van de tekst, en meer in het bijzonder over lexicale en syntactische ambiguïteiten, door de schrijver zelf gegenereerd zijn. Auteursomgevingen, met name meertalige, kunnen hier belangrijke ondersteuning bieden.

LITERATUUR

- Berkel, B. van, *Een algoritme voor spel- en typefoutcorrectie gebaseerd op grafeem-foneemomzetting*, Doctoraalscriptie, Katholieke Universiteit Nijmegen, 1986.
- Brown, J.S., 'From cognitive to social ergonomics and beyond', In: D.A. Norman & S.W. Draper (red.) *User centered system design: new perspectives on human-computer interaction*. Hillsdale NJ: Erlbaum, 1986.
- Desain, P., TREE DOCTOR: a software package for graphical manipulation and animation of tree structures, in G. Mulder, G. v.d. Veer (EDS), *Human-computer interaction: Psychonomic aspects?* Berlin: Springer 1987.
- Fraser, 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 Systems Journal*, 1982, 21, 305-326.

- Kamin, J., *Mastering ThinkTank on the IBM PC*. Berkeley CA: Sybex, 1986.
- Kempen, G., *Natuurlijke taal en kunstmatige intelligentie*. Groningen: Wolters-Noordhoff, 1987.
- Kempen, G. & E. Hoenkamp, 'An Incremental Procedural Grammar for sentence formulation', *Cognitive Science*, 1987, 12, 201-258.
- Kempen, G., L. Konst & K. de Smedt, 'Taaltechnologie voor het Nederlands: vorderingen bij de bouw van een Nederlandstalig dialoog- en auteursysteem', *Informatie*, 1984, 26, 878-881.
- Naffah, N., G. Kempen, J. Rohmer, L. Steels, D. Tschritzis & G. White, 'Intelligent workstation in the office. State of the art and future perspectives', In: J. Roukens & J.F. Renuart (red.), *ESPRIT '84: Status report of ongoing work*. Amsterdam: North-Holland, 1985.
- Shneiderman, B., 'The future of interactive systems and the emergence of direct manipulation', *Behavior and Information Technology*, 1982, 1, 237-256.
- De Smedt, K., 'Object-oriented programming in FLAVORS and COMMONORBIT', In: R. Hawley (red.), *Artificial Intelligence Programming Environments*. London: Ellis Horwood, 1987.
- Smedt, K. de & G. Kempen, 'Coordination, self-correction, and incremental sentence production', In: G. Kempen (red.), *Natural language generation: New results in Artificial Intelligence, psychology, and linguistics*. Dordrecht/Boston: Kluwer Academic Publishers, 1987.
- Deze tekst is een licht bewerkte vertaling (door Drs. Fiency Pijs) van het artikel 'Author Environments: Fifth Generation Text Processors', dat is verschenen in *ESPRIT '86: Results and Achievements* (redactie Directorate General XIII; Amsterdam: North-Holland, 1987).
- Gerard Kempen, Peter Desain en Koenraad de Smedt zijn verbonden aan het Psychologisch Laboratorium, Katholieke Universiteit Nijmegen, Montessorilaan 3, 6525 HR Nijmegen.*
- Gert Anbeek is werkzaam bij Océ-Nederland, St. Urbanusweg 43, 5914 CA Venlo.*
- Leo Konst is werkzaam bij Linguistic Systems, Groenestraat 294, 6531 JC Nijmegen.*