



Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

L.G. Bouma, J. Bruijning, J.C. van Vliet

Documentverwerking

Afdeling Informatica

Notitie CS-N8504

April

Document processing

Bibliotheek
Centrum voor Wiskunde en Informatica
Amsterdam

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

69K82
69H41

Documentverwerking

L.G. Bouma, J. Bruijning

Facultaire Vakgroep Informatica, Universiteit van Amsterdam

J.C. van Vliet

Centrum voor Wiskunde en Informatica, Amsterdam

De verwerking van documenten via beeldschermen en gebruik makend van computers is sterk in opmars. Systemen voor het verwerken van documenten variëren van eenvoudige programma's voor de opmaak van brieven en memo's tot zeer geavanceerde systemen voor de productie van boeken met een complexe inhoud. In dit artikel wordt een overzicht geboden van de belangrijke ontwikkelingen op dit terrein. Hierbij wordt tevens een blik in het kantoor van de toekomst geworpen, waar geavanceerde werkstations het mogelijk maken interactief teksten (inclusief formules, tabellen, tekeningen e.d.) te manipuleren en verwerken.

1980 Mathematics Subject Classification: 68K05

1983 CR Categories: H.4.1 [Office automation]: word processing

Noot: dit artikel is elders ter publicatie aangeboden.

Het via schrift overbrengen van informatie is niet van vandaag of gisteren. In Zuid-Frankrijk zijn rotstekeningen gevonden die van 30.000 v. Chr. dateren. Van latere datum is het beeldschrift, waarin men gebruik maakt van tekeningetjes (pictogrammen). Beeldschrift wordt tegenwoordig gebruikt op verkeersborden, bij bewegwijzering in stations, ed. Het hiërogliefenschrift is een soort tussenvorm. Hierin komen naast pictogrammen ook lettergreeptekens en fonetische hulptekens voor.

De laatste stap in de ontwikkeling is het fonetisch schrift dat *alfabetisch* of *syllabisch* kan zijn. Ons schrift is alfabetisch: iedere letter stelt één spraakklank voor. Het Japanse katakana is syllabisch.

Het hiërogliefenschrift heeft, via het Sinaï-schrift, tot het Fenicisch geleid. Hierin kwamen 22 medeklinkers voor. Het Fenicisch is door de Grieken tot een compleet alfabet gemaakt. Aanvankelijk werd het van rechts naar links geschreven, later om en om van links naar rechts en van rechts naar links, zoals de boer zijn land ploegt. Tenslotte hebben de Romeinen het alfabet verder ontwikkeld.

Het Romeinse alfabet telde aanvankelijk een 20-tal letters, waaraan in een later stadium de G, Y en Z werden toegevoegd. In de 10^e eeuw trad een splitsing tussen U en V op, terwijl pas in de 17^e eeuw de I en J uiteengingen en de W ons huidige alfabet completeerde.

Oorspronkelijk gebruikte men voor de cijfers een additief systeem, zoals dat van de Romeinse cijfers. Rond 500 n. Chr. werd de 0 'uitgevonden' in India en kon er een zuiver positiestelsel ontstaan waarin de waarde van een cijfer afhangt van zijn positie in het getal. De Indische cijfers zijn rond het jaar 1000 via de Arabieren in Europa geïntroduceerd.

Voor meer uitvoerige informatie over het ontstaan en de ontwikkeling van ons schrift raadplege men bij voorbeeld [Treebus83], [Barthel72], of [Diringer77].

Gedrukte teksten hebben eveneens een lange geschiedenis. De uitvinding van de boekdrukkunst wordt algemeen toegeschreven aan Johann Gutenberg (circa 1440). Hij maakte losse drukletters van metaal. Omdat gedrukte tekst zoveel mogelijk op geschreven tekst moest lijken had hij verschillende versies van elke letter, alsmede een aantal combinaties (ligaturen). Gutenberg gebruikte ongeveer 300 karakters.

Er werden al veel eerder afdrucken gemaakt van in hout uitgesneden figuren. In Europa werd deze techniek vanaf de 6^e eeuw gebruikt voor het bedrukken van textiel. In China werd al rond het jaar 1000 gedrukt vanaf houten blokken en losse uit klei gebakken symbolen. Deze uitvinding was echter niet erg belangrijk omdat men niet over een alfabet beschikte en dus heel veel vormen moest bewaren.

Notitie CS-N8504

Centrum voor Wiskunde en Informatica

Postbus 4079, 1009 AB Amsterdam

De oudste manier van zetten is het handzetten. Staande achter een 'bok', een houten meubel met schuin bovenblad, pakte de handzetter met zijn ene hand loden letters uit zijn letterkast en plaatste deze in een rekje in zijn andere hand. De regels werden uitgevuld met wit om ze alle dezelfde lengte te geven, en vervolgens in een mal bijeengebonden. Dit handzetten is een bijna uitgestorven vak en de letterkasten vindt men enkel nog als wandversiering in de huiskamer.

In de tweede helft van de 19^e eeuw werd het machinaal zetten ontwikkeld. Bij de *Linotype* worden via aanslagen op een toetsenbord matrijzen (dunne koperen plaatjes met een verdiept letterbeeld) verzameld. Bij een volle regel wordt er in een gietvorm lood tegenaan gespoten en ontstaat er een loden regel. Bij correctie moet steeds de hele regel opnieuw gezet worden. Bij de *Monotype*-zetmachine wordt de gietmachine op een vergelijkbare manier gestuurd; hier echter blijft het gietstel uit losse lettertjes bestaan zodat er iets flexibeler correctiemogelijkheden zijn.

Na de tweede wereldoorlog ontstond het fotozetten, waarbij tekens fotografisch op film worden vastgelegd. Huidige fotografische zetmachines worden veelal door computers gestuurd. Hierbij wordt tekst via een beeldscherm ingevoerd en opgeslagen. Veelal vindt het echte formatteren in een aparte fase plaats. Tijdens dit proces wordt de tekst in alinea's en pagina's gegoten, welke vervolgens naar de fotozetmachine gestuurd worden. Het formatteren van de tekst wordt veelal gestuurd door in de tekst opgenomen commando's.

Sinds de opkomst van de boekdrukkunst zijn er vele fonts - verzamelingen karakters met een bepaald uiterlijk en in een bepaalde stijl - ontwikkeld. Deze ontwikkeling is voor een deel technologie-gestuurd: bij het zetten uit lood gelden andere eisen voor de precieze vorm van karakters dan bij fotografisch zetten. Bij de weergave van tekst op een beeldscherm gelden nog weer andere regels.

Heel vaak nog is de weergave van tekst op een beeldscherm typografisch gezien verre van volmaakt. Voor een belangrijk deel wordt dit veroorzaakt door de beperkte resolutie van beeldschermen. Huidige grafische beeldschermen hebben veelal een resolutie van 70-100 bits (*pixels*) per inch. Fotozetmachines hebben een resolutie in de orde van 1500 pixels per inch. Hierbij komt nog dat inkt vloeit, zodat de contouren van karakters vanzelf een vloeiende vorm krijgen. Er moeten dan ook speciale technieken gebruikt worden bij de ontwikkeling van ook op beeldschermen goed leesbare fonts [Bigelow83, 85]. Een van de mogelijkheden is om per pixel op het beeldscherm naast de mogelijkheden "wit" en "zwart" ook een aantal grijswaarden te bieden.

Het ontwerpen van typefonts is een vak apart. Ook hierbij kan de computer ingeschakeld worden. IMP [Carter84], LIP [Flowers84] en Metafont [Knuth79] zijn state-of-the-art voorbeelden van (interactieve) systemen voor het ontwerpen van typefonts.

Voor meer uitvoerige informatie over typografie en de verschillende methoden van zetten kan men te rade gaan bij o.a. [Treebus83] en [Groenendaal75].

In dit artikel zal een overzicht gegeven worden van de verschillende systemen die zoal ontwikkeld zijn voor het elektronisch verwerken van documenten. In de volgende sectie wordt een tweedeling voorgesteld in systemen voor het verwerken van teksten waarin gebruik wordt gemaakt van eenvoudige grafische mogelijkheden, zoals brieven en memo's, en systemen voor het verwerken van complex materiaal. In secties 2 en 3 worden enkele belangrijke voorbeelden uit beide categorieën besproken.

Er is een groeiende belangstelling in interactieve systemen voor het verwerken van documenten. In het kantoor van de toekomst zal veel uitwisseling van informatie via beeldschermen verlopen, en men mag verwachten dat er systemen zullen komen die een steeds betere weergave van documenten met een gevarieerde inhoud kunnen bewerkstelligen. Op deze ontwikkeling wordt in sectie 4 nader ingegaan.

Een uitgebreid overzicht van verschillende systemen voor het verwerken van documenten is ook te vinden in [Furuta82]. In [Sigplan81] en [Miller84] zijn artikelen gebundeld die gepresenteerd zijn op conferenties over dit onderwerp. In [Nievergelt82] tenslotte zijn een aantal eerder verschenen artikelen gebundeld.

1. Word processing vs document processing

Een typemachine is een uitstekend medium om eenvoudige documenten als brieven of rapporten met platte tekst te produceren. Het is echter lastig om bij deze aanpak niet-triviale correcties toe te passen. Wanneer men gebruik maakt van een computer wordt dit een stuk eenvoudiger. Met behulp van een teksteditor wordt een brief opgemaakt welke vervolgens op een aan de computer gekoppelde typemachine kan worden afgedrukt.

Bij deze aanpak wordt het ook mogelijk eenvoudige opmaak van tekst te automatiseren. De tekst kan automatisch uitgevuld worden, zodat er zoveel mogelijk woorden op elke regel staan. Men kan de tekst uitlijnen, hetgeen resulteert in een strakke rechterkantlijn. Er kan een routine ingebouwd worden die tracht woorden af te breken welke niet meer op de regel passen [Van Vliet72], [Boot82].

Er zijn een groot aantal systemen op de markt voor dit soort toepassingen. In een aantal gevallen betreft dit software-pakketten voor algemene microcomputers, in een aantal gevallen ook betreft het een speciaal op tekstverwerking gerichte configuratie. In het laatste geval heeft men vaak de beschikking over een toetsenbord met aan de toepassing aangepaste functietoetsen.

Bij veel van deze systemen zijn ook de commando's van de editor toegespitst op het gebruik. Commando's als "ga naar de volgende alinea" komt men veelvuldig tegen.

De uitvoer vindt in het algemeen plaats via een typemachine of matrixprinter. In het laatste geval heeft men vaak iets meer mogelijkheden; tekst kan hier bij voorbeeld ook cursief of vet afgedrukt worden, al is het resultaat typografisch gezien niet altijd even fraai.

Deze klasse van systemen wordt in dit artikel met de term 'word processing' aangeduid, en in sectie 2 nader besproken.

Aan het andere eind van het spectrum vindt men de systemen welke uitvoer produceren voor een fotozetmachine of laserprinter. Deze bieden de mogelijkheid tekst te zetten in verschillende fonts en in uiteenlopende groottes. Er zijn vaak ook fonts beschikbaar met speciale symbolen, zoals die welke in de wiskunde worden gebruikt. Zij bieden tenslotte soms grafische mogelijkheden, zodat er tekeningen geproduceerd kunnen worden. Tezamen biedt dit de mogelijkheid complexe documenten te produceren.

De bijbehorende formatterprogramma's hebben vaak uitgebreidere mogelijkheden voor tekstopmaak, zoals:

- het automatisch genereren van kopregels, waarin bij voorbeeld de titel van een hoofdstuk of sectie op elke pagina wordt herhaald;
- het automatisch plaatsen van voetnoten onderaan de pagina;
- het automatisch genereren van een index of inhoudsopgave;
- het oplossen van kruisreferenties, zoals "zie pagina \$A", waarbij het systeem de tekst "\$A" vervangt door het juiste paginanummer;
- het, binnen zekere marges, schuiven met plaatjes of tabellen, zodat deze niet over paginagrenzen geplaatst worden.

In sectie 3 wordt nader ingegaan op enkele systemen uit deze categorie en wordt door middel van voorbeelden een indruk gegeven van hun mogelijkheden.

Ook bij deze systemen wordt de invoer via teksteditoren aangemaakt. De meeste beeldschermen bieden echter niet voldoende mogelijkheden om een getrouw beeld van de uitvoer te geven. Het formatterproces is hier ook nogal gecompliceerd en tijdrovend zodat dit in het algemeen off-line geschiedt.

We zullen in dit artikel niet nader ingaan op de eigenschappen en mogelijkheden van de verschillende teksteditoren. In [Meyrowitz82] wordt een uitstekend overzicht geboden van deze materie.

In tabel I zijn de belangrijkste verschillen tussen beide categorieën systemen nog eens samengevat. Het spreekt vanzelf dat de grens tussen beide typen niet scherp te trekken is. (Er zijn ook betrekkelijk eenvoudige en goedkope systemen die voorzien zijn van een kleine fotozetmachine.)

Toch achten wij het belangrijk dit onderscheid te maken. Veel van de tegenwoordige systemen voor het verwerken van documenten bieden de gebruiker slechts beperkte mogelijkheden. Er zijn ook professionele systemen die de gebruiker de mogelijkheid bieden complexe documenten op zeer fraaie wijze te verwerken. Wij verwachten dat hier een zekere verschuiving op zal treden, zodat men in de toekomst ook op algemeen gangbare werkstations teksten op een flexibele en aantrekkelijke wijze kan verwerken.

Word processing	Document processing
Interactief	Batch
Eenvoudige documenten	Complexe documenten
Matrixprinter	Fotozetmachine/laserprinter

Tabel I: Globale classificatie van systemen voor documentverwerking.

2. Word processing

In deze paragraaf zullen we het begrip "word processing" meer inhoud geven. Karakteristieke eigenschappen voor de betrokken systemen zijn o.a.:

- de te bewerken tekst wordt full-screen gepresenteerd; het scherm is als het ware een venster op een deel van het gehele document;
- de gebruiker beschikt over een cursor waarmee in principe iedere plaats binnen de tekst bereikbaar is;
- het systeem is gericht op het bewerken van platte tekst (geen grafieken of plaatjes);
- het systeem bevat (vrijwel) geen kennis over verschillende typefonts;
- het systeem heeft geen kennis van de opbouw van ingewikkelder documenten zoals bijvoorbeeld het manuscript van een boek (inhoudsopgave, index, automatische hoofdstuknummering);
- het systeem is gericht op het werken met eenvoudige uitvoerapparatuur (daisy-wheel printers, matrix printers; geen geavanceerde laserprinters of fotozetmachines).

De systemen die wij op het oog hebben zijn al meerdere jaren wijdverspreid, vooral in kantooromgevingen, maar ook steeds meer in de huiselijke sfeer. We onderscheiden een tweetal categorieën: die van de word processors op personal computers, en die van de "dedicated" word processors. In de volgende twee secties worden enkele vertegenwoordigers van beide categorieën genoemd. Wij streven in dit kader uitdrukkelijk niet naar het geven van een marktoverzicht, en pretenderen niet dat de door ons genoemde systemen noodzakelijkerwijs de beste in hun categorie zijn; wel dat zij karakteristiek zijn voor een hele familie.

2.1. Word processors op personal computers

De personal computer is door zijn flexibiliteit al snel ingezet voor word processing. Het lijkt niet overdreven te stellen dat voor velen de mogelijkheid tot word processing de belangrijkste drijfveer is voor de aanschaf van een dergelijk apparaat. Er is weinig grond voor de veronderstelling dat een dedicated word processor beter werk zou leveren dan een personal computer met een goed programma.

We zullen in deze paragraaf aandacht besteden aan een van de bekendste word processors die op personal computers beschikbaar is: WordStar van MicroPro. WordStar is in een vrij vroeg stadium op de markt gekomen. In de loop der jaren zijn er faciliteiten bijgekomen voor gepersonaliseerde standaardbrieven ("mail merge"), toegang tot gegevensbestanden, rekenfaciliteiten ("spread sheet"), enz.

WordStar is nu verkrijgbaar voor vrijwel alle CP/M computers, evenals voor MS-DOS. (CP/M is vrijwel de industrie-standaard wat betreft bedrijfssystemen voor 8-bits microcomputers. De IBM PC, een 16-bits machine, draait onder MS-DOS). Het pakket is full-screen georiënteerd, maar er wordt niet altijd een getrouw beeld van de uiteindelijke uitvoer op het scherm getoond.

Wanneer het programma wordt opgestart, verschijnt een menuscherm met een aantal opties (figuur 1).

```

                                editing no file

D=create or edit a Document file           H=set Help level
N=create or edit a Non-document file       X=eXit to system
M=Merge-print a file                       P=Print a file
F=File directory off (on)                  Y=delete a file
L=change Logged disk drive                 O=cOpy a file
R=Run a program                           E=rEname a file

DIRECTORY of disk B:
ARTIKEL1.DOC PIP.COM                      VOORBEELD

```

Figuur 1: WordStar options menu.

Dit scherm biedt de gebruiker de mogelijkheid de belangrijkste file-georiënteerde commando's uit te voeren: copy, rename, delete. Hierdoor is het voor een gebruiker die alleen in word processing is geïnteresseerd mogelijk alles wat hiermee samenhangt geheel binnen WordStar af te handelen.

De meest voor de hand liggende optie is "D", het bewerken van een "Document file". Wanneer de gebruiker deze optie kiest, wordt hem gevraagd om een filenaam, waarna het "edit" scherm verschijnt.

```
B:VOORBEELD PAGE 1 LINE 8 COL09          INSERT ON  
L-----!----!----!----!----!----!----!----!----!----!-----R  
Dit is een voorbeeld van een Edit scherm. Op de status regel geheel bovenaan  
staat de huidige plaats van de cursor aangegeven, en of ingetypte tekst  
tussengevoegd wordt (INSERT ON) dan wel bestaande tekst overschrijft (INSERT  
OFF). De tweede regel geeft de linker- en rechter marge en de plaats van de  
tabulatorstops aan.  
Afgesloten alinea's worden in de rechter kantlijn met het teken '<'  
aangegeven. De afgesloten regels worden uitgelijnd door tussenvoegen van  
spaties._
```

Figuur 2: Beeldscherm tijdens een sessie met WordStar.

Op de regel geheel bovenaan het scherm staat de documentnaam, en de plaats van de cursor, uitgedrukt in regelnummer en karakterpositie binnen de regel. Daaronder verschijnt de "ruler line", waarop aangegeven de positie van de linker- en rechter kantlijnstop en de tabulatorposities. Figuur 2 toont een beeldscherm tijdens een sessie met WordStar.

Binnen WordStar kan de cursor naar iedere gewenste positie op het scherm gedirigeerd kan worden (althans binnen dat gedeelte van het scherm dat gereserveerd is voor weergave van het document). Voor commando's, dus ook voor de opdracht tot verplaatsen van de cursor, maakt WordStar op de meeste machines geen gebruik van speciale functietoetsen, maar van een combinatie van de "control" toets **Q** en een normale lettertoets. Zo verplaatsen **E**, **S**, **D** en **X** de cursor respectievelijk een plaats naar boven, naar links, naar rechts, en naar beneden. Lijkt dit op het

eerste gezicht onlogisch en moeilijk te onthouden, een blik op het toetsenbord leert dat hierbij rekening is gehouden met de relatieve positie van de betrokken toetsen. Bovendien is gewoonlijk de "control" toets onder bereik van de linkerpink, zodat de cursorbesturing geheel met de linkerhand kan plaatsvinden. Eenmaal geleerd, blijkt deze opstelling ergonomisch zeer geslaagd.

In normale toestand wordt door de gebruiker ingevoerde tekst op de plaats van de cursor tussen de bestaande tekst ingevoegd. Hierbij wordt zo nodig een woord dat niet meer op de regel past in zijn geheel naar de volgende regel verplaatst. De zojuist afgesloten regel wordt meteen uitgelijnd. Wanneer de gebruiker een regel niet wil uitvullen, dus als hij een nieuwe paragraaf wil beginnen, doet hij dit door op de "Return" toets te drukken. WordStar plaatst de cursor dan aan het begin van de volgende regel. Bij dit alles wordt zonodig het "venster" dat WordStar op het document biedt, aangepast.

WordStar stelt de gebruiker ook in staat sub- en superscripts, een alternatief lettertype, vetgedrukte en onderstreepte tekst te genereren. Hiertoe moeten de betrokken stukken tekst omringd worden met zgn. "control characters". De meeste implementaties van WordStar kunnen geen echte onderstreping of vetgedrukte tekst op het scherm laten zien.

Het aldus opgebouwde document kan onder controle van WordStar worden afgedrukt. In het algemeen zullen dan wel alle aangebrachte attributen, zoals onderstreping, zichtbaar worden.

WordStar kent een groot aantal printers en tracht op zo efficiënt mogelijke wijze het gewenste effect te bereiken. Bij printers die in staat zijn tot proportionele spatiëring, wordt hiervan gebruik gemaakt bij het uitlijnen.

WordStar kent nog een andere manier om de layout van het gedrukte document te beïnvloeden. Dit gebeurt via in de tekst ingebedde commando's die aan het begin van een regel staan en met een punt beginnen. Men kan er het aantal regels op een pagina mee sturen, een pagina-overgang forceren, een lopende kopregel mee definiëren, de bladzijnummering mee sturen, enz. (zie tabel II). Deze commando's worden op het beeldscherm niet geëffectueerd. Dit is deels de prijs die men betaalt voor terminal- en printeronafhankelijkheid, en deels een gevolg van de relatief simpele hardware waarop het systeem draait. Desondanks kan WordStar een gebruikersvriendelijk product genoemd worden.

.pl N	pagina lengte is N regels
.pa	forceer pagina overgang
.he '...'	definieer kopregel met tekst '...'
.pn N	begin pagina nummering bij N
.op	nummer de pagina's niet
.cw N	karakter breedte is N dots (voor daisy wheel printers)
.lh N	regelafstand is N dots (voor daisy wheel printers)

Tabel II: Enkele layout commando's.

Omvangrijker systemen als van Wang en IBM leveren additionele faciliteiten, vooral op het punt van de administratie van alle documenten, iets wat voor een omgeving met meerdere gebruikers van groot belang kan zijn.

2.2. Dedicated word processors

Daar het gebruikersinterface van specifiek op word processing gerichte configuraties niet principieel van dat van WordStar verschilt, volstaan we met het noemen van enkele punten.

Wang word processors zijn in vele kantoren te vinden. Het zijn in wezen microcomputers die speciaal voor word processing zijn uitgerust. Kenmerkende aspecten zijn:

- uiterlijk van een klassieke terminal met scherm en toetsenbord, voorzien enkele speciale toetsen;
- selectie van stukken tekst d.m.v. highlighting;

- functietoetsen voor copy, move, delete, enz.;
- elke opdracht moet bevestigd worden;
- menu-gestuurde selectie van files en van faciliteiten zoals backup;
- een uitgebreide index waarin informatie over de files is opgeslagen, zoals tijdstip van creatie, tijdstip van laatste wijziging, auteur, typist(e), grootte.

Visietekst is IBM's versie van de word processor. Enkele bijzondere karakteristieken zijn:

- gegevens kunnen vanuit bestanden worden ingelezen en in de tekst geïncorporeerd;
- er is een beperkte rekenfaciliteit, waarmee bij voorbeeld het totaal van een kolom cijfers kan worden berekend en in het document geplaatst;
- het systeem kan de spelling controleren.

Deze systemen zijn exemplarisch voor een veel grotere familie; vele leveranciers hebben hun eigen variatie op dit thema.

3. Document processing

In deze sectie zullen vier systemen worden besproken voor het verwerken van documenten met een complexe inhoud. De uitvoer van deze systemen wordt in het algemeen verwerkt door een fotozetmachine of laserprinter.

Troff en T_EX zijn bekende batchgeoriënteerde systemen. Beide bieden een uitgebreid scala aan mogelijkheden; men moet een expert in het gebruik zijn om ten volle profijt van deze mogelijkheden te trekken. Door echter gebruik te maken van de geboden macro-faciliteiten kan de opmaak van documenten met een standaard structuur sterk vereenvoudigd worden.

Deze aanpak is in Scribe verder doorgevoerd. Scribe maakt gebruik van een databank waarin voor elk type document is vastgelegd wat de structuur is en hoe deze structuur tot een specifieke layout leidt. De gebruiker hoeft in zijn invoertekst nu enkel deze structuur aan te geven.

Standaardisatie op het gebied van tekstverwerking vindt plaats in ISO-verband. Ook in het daar gekozen systeem - SGML genaamd - geeft de gebruiker in de invoer de structuur van het document aan. Belangrijk is dat SGML faciliteiten biedt om de syntactische vorm van opmaakcommando's te definiëren, zodat aansluiting op reeds bestaande systemen mogelijk wordt.

3.1. Troff

Onder het UNIX* bedrijfssysteem zijn een aantal programma's beschikbaar voor het verwerken van documenten. De centrale component hieruit is het programma Troff, dat omstreeks 1973 is ontwikkeld [Ossanna76]. Troff is een tekstformatter die uitvoer levert voor een fotozetmachine. De meest recente versie staat bekend onder de naam Ditroff [Kernighan82a]; de uitvoer van Ditroff is apparaat-onafhankelijk, zodat koppeling met verschillende fotozetmachines relatief eenvoudig is. Er is ook een, grotendeels met Troff compatibel, programma, genaamd Nroff, dat uitvoer produceert voor een regeldrukker of typemachine. In het nu volgende zullen wij de discussie tot Troff beperken.

De invoer van Troff is een standaard UNIX tekstfile. Regels in de invoer die beginnen met een "." worden opgevat als speciale commando's voor de formatter. In tabel III zijn een aantal van deze commando's met hun betekenis gegeven. De invoerregels die niet met een "." beginnen worden, onder besturing van de geldende Troff-commando's, naar de uitvoer gecopieerd.

De basis-commando's uit Troff zijn betrekkelijk primitief. De gebruiker kan typefonts selecteren, de hoogte en breedte van pagina's vaststellen, laag-niveau layout-aanwijzingen geven, e.d. Troff biedt daarnaast de mogelijkheid reeksen layout-aanwijzingen in een macro te combineren, zodat vervolgens deze macro's als commando gebruikt kunnen worden. Men kan op deze wijze een serie macro's definiëren die tezamen de stijl van een document bepalen. Het standaard macro-pakket staat bekend onder de naam *ms* [Lesk78]. In tabel IV zijn enkele veel gebruikte *ms*-macro's

* UNIX is een handelsmerk van Bell Laboratories.

Commando	Initiële waarde	Betekenis
.ps N	10 punts	Verander de lettergrootte in N punten.
.vs N	12 punten	Verander de verticale afstand tussen opeenvolgende regels. Deze afstand kan uitgedrukt worden in punten, inches, en centimeters.
.ft F	Roman	Ga over op het font met code F.
.pl N	11 inch	Zet de paginalengte op N (centimeter, inches, punten).
.ll N	6.5 inch	Zet de regellengte.
.bp	1	Ga over op een nieuwe pagina. De huidige pagina wordt afgesloten. Het (intern bijgehouden) paginanummer wordt met 1 opgehoogd.
.ne N	-	Geeft aan dat er een verticale ruimte ter hoogte N nodig is. Indien deze ruimte op de huidige pagina niet meer aanwezig is wordt er een overgang naar een nieuwe pagina gegenereerd.
.nf	opvullen	De tekst volgend op dit commando wordt rechtstreeks naar de uitvoer gecopieerd. M.a.w., regelovergangen in de invoer leiden ook tot regelovergangen in de uitvoer.
.fi	opvullen	Vanaf dit punt wordt weer zoveel mogelijk tekst op elke regel geplaatst.
.na	uitlijnen	Vanaf dit punt worden de regels niet uitgelijnd. M.a.w., de rechterkantlijn wordt rafelig.
.ce N	-	De volgende N uitvoerregels worden gecentreerd op de pagina geplaatst.

Tabel III: Enige Troff-commando's met hun betekenis.

Commando	Betekenis
.TL	Hierna volgt de titel van het document.
.AU	Hierna volgen de namen van de auteurs.
.AB	Begin abstract.
.AE	Einde abstract.
.NH N	Begin een nieuwe sectie op niveau N.
.PP	Begin een nieuwe paragraaf.
.FS	Nu volgt een voetnoot.

Tabel IV: Enige ms-commando's met hun betekenis.

met hun betekenis opgenomen. Figuur 3 bevat een voorbeeld van het gebruik van dit pakket.

Deze vorm van gebruik is natuurlijk verre te prefereren boven die waarin uitgegaan wordt van de primitieve commando's. Men kan éénmaal een verzameling macro's definiëren, en deze vervolgens voor een reeks documenten, zoals een rapport-serie, gebruiken.

Troff stelt ons dus in staat op flexibele wijze platte tekst te verwerken. Daarnaast zijn er onder UNIX aparte programma's beschikbaar voor het verwerken van formules (EQN), tabellen (TBL) en plaatjes (PIC). Deze programma's werken alle als preprocessor voor Troff. Een programma als EQN [Kernighan75] filtert uit een invoertekst die delen waarin formules staan beschreven en bewerkt deze tot (laag-niveau en voor de gebruiker onleesbare) tekst welke vervolgens door Troff verder verwerkt wordt.

In figuur 4 is een voorbeeld gegeven van een tekst waarin formules voorkomen. Een formule wordt omgeven door de commando's .EQ en .EN. EQN verwerkt alleen de tekst tussen deze twee commando's, en copieert de rest van de invoer zonder meer naar de uitvoer. Tussen de commando's .EQ en .EN staat tekst die qua structuur nogal afwijkt van de normale Troff-invoer. Formules worden beschreven in een taal die vrij natuurlijk overkomt. Het EQN-programma reageert op woorden als *sum*, *int*, *over*, enz. De overige tekst wordt cursief gezet, tenzij het om woorden gaat met een standaard wiskundige betekenis, zoals *sin*, *log*, *max*.

Op een vergelijkbare manier reageert TBL [Lesk79] op invoer tussen de commando's .TS en .TE. Voor elke tabel geeft de gebruiker achtereenvolgens de opties, het formaat, en de gegevens die in de tabel moeten worden geplaatst. Bij de opties kan o.a. worden opgegeven of de tabel gecentreerd op de pagina moet worden geplaatst, en of er lijnen rond de gehele tabel getekend

Invoertekst:

.TL

Software engineering

.AU

J.C. van Vliet

.AB

In dit artikel wordt beargumenteerd dat Software Engineering als verplicht vak op Universiteiten en Hogescholen ingevoerd zou moeten worden.

.AE

.NH

INLEIDING

.LP

Programmatuur wordt vaak te laat opgeleverd, voldoet niet aan de specificaties, en bevat bovendien fouten.

.NH 2

Waarom Software Engineering ook een leuk vak is

.LP

In tegenstelling tot Functietheorie is Software Engineering een vak waarbij men ...

Uitvoer:

Software engineering

J.C. van Vliet

ABSTRACT

In dit artikel wordt beargumenteerd dat Software Engineering als verplicht vak op Universiteiten en Hogescholen ingevoerd zou moeten worden.

1. INLEIDING

Programmatuur wordt vaak te laat opgeleverd, voldoet niet aan de specificaties, en bevat bovendien fouten.

1.1. Waarom Software Engineering ook een leuk vak is

In tegenstelling tot Functietheorie is Software Engineering een vak waarbij men ...

Figuur 3: Het gebruik van het ms-pakket.

moeten worden. In de formaat-sectie wordt de layout van de tabel gespecificeerd. Elke regel in deze sectie geeft de layout van 1 uitvoerregel, met uitzondering van de laatste, die de layout van alle volgende regels aangeeft. De gebruiker geeft in de layout-sectie per kolom indicaties v.w.b. het formaat, zoals:

- moet de tekst links of rechts uitgelijnd worden, of gecentreerd (code l, r of c);
- loopt de tekst van de vorige kolom door in deze kolom (code s);
- hoeveel ruimte moet er tussen kolommen opengelaten worden;
- hoe breed moet een kolom (minimaal) worden;
- moeten kolommen gelijke breedte krijgen.

Vervolgens wordt de inhoud van de tabel gegeven. In figuur 5 is een voorbeeld van een eenvoudige tabel gegeven. De verschillende kolommen worden door een speciaal symbool gescheiden; in figuur 5 is dit het symbool "@".

Het programma PIC [Kernighan82b] wordt gebruikt voor het produceren van plaatjes. De invoer voor PIC lijkt qua structuur sterk op die van EQN. De gebruiker specificeert in een betrekkelijk natuurlijk aandoende taal de structuur van het plaatje. Hem staan hierbij standaard

Invoertekst:

```
Na integratie verkrijgen we
.EQ
W(t) = 1 - exp ( - int to t p ( alpha ) d alpha )
.EN
Stel p(t) = at.
De voortgang wordt dan gegeven door een Rayleigh-verdeling:
.EQ
dW over dt = ate sup { -at sup 2 / 2 }
.EN
```

Uitvoer:

Na integratie verkrijgen we

$$W(t) = 1 - \exp(-\int p(\alpha) d\alpha)$$

Stel p(t) = at. De voortgang wordt dan gegeven door een Rayleigh-verdeling:

$$\frac{dW}{dt} = ate^{-at^2/2}$$

Figuur 4: Het gebruik van EQN voor formules.

Invoertekst:

```
.TS
center, box, tab(@);
cfB s s
c c c
^c c
l n n.
Omvang programma's
Naam@Regels@Object bytes
@source code@(tekst + data)

Troff@8681@73136
EQN@1821@34164
TBL@2581@39936
PIC@3760@83968
.TE
```

Uitvoer:

Naam	Omvang programma's	
	Regels source code	Object bytes (tekst + data)
Troff	8681	73136
EQN	1821	34164
TBL	2581	39936
PIC	3760	83968

Figuur 5: Voorbeeld van het gebruik van TBL (Bron: [Kernighan84]).

componenten ter beschikking als doosjes, cirkels, ellipsen en lijnen, met behulp waarvan het plaatje opgebouwd wordt. In figuur 6 is een eenvoudig voorbeeld van het gebruik van PIC gegeven.

De hierboven summier besproken programma's bieden veel meer mogelijkheden dan in dit korte bestek is aangegeven. Ze bieden zelfs zoveel dat er al gauw een expert nodig is. Dit geldt zeker voor formules, plaatjes, e.d., waar aan de invoer soms moeilijk te zien is hoe de uitvoer eruit zal zien.

Een tweede probleem is dat de verschillende programma's niets van elkaar weten. Zo gaat EQN er standaard van uit dat de tekst in een 10-punts letter wordt gezet. Op soortgelijke wijze weet het programma PIC niets van de grootte van de eventuele tekst die in plaatjes wordt geplaatst. Dit kan leiden tot resultaten als in figuur 7 weergegeven.

Invoertekst:

.PS

B0: box ht 0.4i wid 1.5i "bepaal startpunt" "voor iteratie"

arrow down 0.5i from B0.s

box ht 0.4 wid 1.2 "x: = f(i)"

arrow down 0.5i

BI: box invis "x > eps"

line from BI.n to BI.e

line from BI.e to BI.s

line from BI.s to BI.w

line from BI.w to BI.n

arrow down 0.5 from BI.s "nee" ljust

box ht 0.4i wid 1.2i "nulpunt" "gevonden"

arrow right 1.0i from BI.e "ja" above

B: box ht 0.4i wid 1.2i "i: = next(i)"

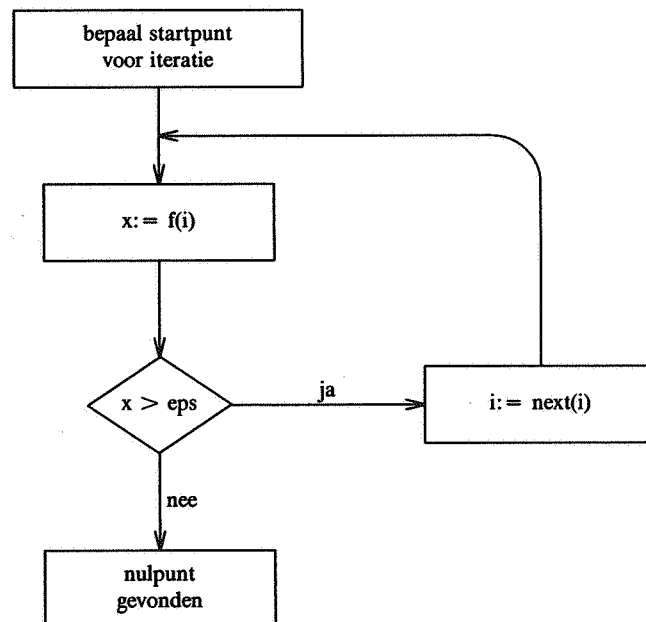
line up 0.95i from B.n

arc

arrow left 1.725i

.PE

Uitvoer:



Figuur 6: Voorbeeld van het gebruik van PIC.

Troff is al een tiental jaren oud. In de loop der tijd zijn de diverse preprocessors toegevoegd en dit heeft de structuur van het geheel geen goed gedaan. Het systeem biedt de ervaren gebruiker vele mogelijkheden. Minder ervaren gebruikers kunnen zich beperken tot het gebruik van macropakketten en zo met weinig moeite gezette documenten van heel behoorlijke kwaliteit produceren. De geïnteresseerde lezer wordt voor verdere details verwezen naar [Akkerhuis84] en de hierboven aangehaalde literatuur.

Invoertekst:

```
.nr PS 16
.nr VS 20
Deze tekst wordt groot gezet, maar de formule
.EQ
FP = sum from i TC sub i * log sub 2 TC sub i
.EN
helaas niet.
```

Uitvoer:

Deze tekst wordt groot gezet, maar de formule

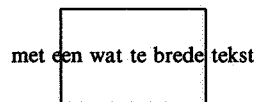
$$FP = \sum_i TC_i * \log_2 TC_i$$

helaas niet.

Invoertekst:

```
.PS
box "met een wat te brede tekst"
.PE
```

Uitvoer:



Figuur 7: Twee onverwachte resultaten.

3.2. T_EX

T_EX [Knuth79, 84] is in de tweede helft van de 70-er jaren ontwikkeld door D.E. Knuth, auteur van het monumentale werk "The Art of Computer Programming", waarvan de eerste 3 delen zijn verschenen. Het verhaal gaat dat, toen er een revisie van deel 2 moest verschijnen, de uitgever meedeelde dat dit niet meer zo fraai kon worden uitgevoerd als de eerste druk, omdat bestaande door een computer ondersteunde programmatuur voor het zetten van teksten daartoe niet in staat zou zijn. Knuth wilde dit niet accepteren en ontwikkelde T_EX om het tegendeel aan te tonen.

T_EX staat voor TAU EPSILON CHI en wordt uitgesproken als 'tech'; de letters $\tau\epsilon\chi$ zijn de eerste letters van het Griekse woord dat kunde betekent; deze stam komt ook in woorden als 'technologie' terug.

T_EX is, in Knuth's woorden, bedoeld om *mooie* boeken te produceren. In T_EX is veel aandacht besteed aan technieken om wiskundig en tabulair materiaal te verwerken. In tegenstelling tot Troff geschiedt dit niet via preprocessors. Een van de in het oog lopende kenmerken van T_EX is de fraaie wijze waarop gewone platte tekst wordt behandeld.

Een centrale rol in T_EX speelt de 'box'. Een 'box' verwijst naar iets dat gezet kan worden. Dit kan een los karakter zijn, een heel woord, een wiskundige formule, o.i.d. Deze doosjes worden naast en onder elkaar geplaatst om zo een regel, alinea of pagina te vullen. Bij dit opvullen spelen alleen de hoogte en breedte van deze doosjes een rol, en niet de inhoud ervan. Doosjes zijn met elkaar verbonden door 'lijm'; hier zit een zekere speling in zodat gezocht kan worden naar een optimale opmaak.

Voor platte tekst kan men voor deze doosjes de woorden van een alinea nemen, en voor de lijm de variabele spaties tussen de woorden. Tussen opeenvolgende woorden is er een zekere nominale witruimte. Deze kan iets groter of kleiner gemaakt worden, al naar gelang dit beter

uitkomt bij het invullen van de regels. Bij voorkeur wordt zo weinig mogelijk van de nominale witruimte tussen woorden afgeweken. Hoe meer hiervan wordt afgeweken, hoe slechter de tekst gezet is.

Men kan dit ook in termen van 'straf' zien. De straf voor een op een bepaalde manier gezette alinea is dan een maat voor de mate waarin voor deze alinea afgeweken is van de nominale witruimte tussen woorden. De beste manier om een stuk tekst te zetten is dan die verdeling over regels die de laagste totale straf oplevert. Knuth heeft hiervoor een heel fraaie algoritme ontwikkeld [Knuth81]. Deze algoritme houdt nog met een aantal andere zaken rekening, zoals:

- het afbreken van woorden moet zoveel mogelijk worden voorkomen;
- de laatste regel van een alinea moet bij voorkeur uit meer dan 1 woord bestaan;
- op opeenvolgende regels moet liefst ongeveer evenveel wit tussen de woorden staan (een regel met veel wit tussen woorden, gevolgd door een regel met erg weinig wit tussen woorden, is storend voor het oog).

Dit soort zaken kan ook netjes in termen van straffen worden geformuleerd en wordt dan verder automatisch door Knuth's algoritme verwerkt.

De manier waarop in $\text{T}_\text{E}\text{X}$ tekst over alinea's wordt verdeeld is fraaier dan de manier waarop dit in Troff gebeurt. In Troff wordt zoveel mogelijk tekst op een regel geplaatst. Is de regel vol dan vindt een overgang naar een nieuwe regel plaats. In $\text{T}_\text{E}\text{X}$ wordt de tekst van een hele alinea bekeken, en dit levert in het algemeen een evenwichtiger geheel op [Brown84].

Net als in Troff maakt men bij $\text{T}_\text{E}\text{X}$ veelal gebruik van een macropakket waarin de standaard structuur van een document is gedefinieerd. Gebruik makend van het $A_M\text{S-}\text{T}_\text{E}\text{X}$ pakket [Spivak83] ziet de invoertekst van figuur 3 er uit als in figuur 8 weergegeven. Hier wordt in de eerste regel de naam van het te gebruiken pakket gegeven, terwijl in de tweede regel voor een specifiek formaat binnen de door het pakket geboden mogelijkheden wordt gekozen. De overige invoer spreekt voor zich. De uitvoer is vergelijkbaar met die van het eerder besproken ms-pakket.

Invoertekst:

```
\input amstex
\documentstyle{amsppt}
\title Software engineering \endtitle
\author J.C. van Vliet \endauthor
\abstract{In dit artikel wordt beargumenteerd dat Software Engineering
als verplicht vak op Universiteiten en Hogescholen ingevoerd
zou moeten worden}
\heading 1. Inleiding \endheading
Programmatuur wordt vaak te laat opgeleverd, voldoet niet aan de
specificaties, en bevat bovendien fouten.
\subheading{1.1. Waarom Software Engineering ook een leuk vak is}
In tegenstelling tot Functietheorie is Software Engineering een vak
waarbij men ...
```

Figuur 8: Voorbeeld van $A_M\text{S-}\text{T}_\text{E}\text{X}$ -invoer.

In $\text{T}_\text{E}\text{X}$ is erg veel aandacht besteed aan mogelijkheden om formules te zetten, en de behandeling van formules is in het systeem geïntegreerd. Formules worden tussen $\$$ -symbolen geplaatst (of, indien ze op een aparte regel moeten worden geplaatst, tussen dubbele $\$$ -symbolen). Er wordt bij de beschrijving van formules gebruik gemaakt van een taal die lijkt op die van EQN. Er zijn geen gereserveerde woorden als in EQN; in plaats daarvan beginnen dit soort symbolen steeds met " $\backslash$ ". Het repertoire aan mogelijkheden is zeer uitgebreid. In figuur 9 is als voorbeeld de $\text{T}_\text{E}\text{X}$ -invoer voor de tekst uit figuur 4 gegeven.

Tenslotte zijn er ook in $\text{T}_\text{E}\text{X}$ mogelijkheden om tabellen te produceren. Hierbij maakt men gebruik van betrekkelijk laag-niveau commando's, hetgeen tot gevolg heeft dat de invoertekst voor tabellen al snel vrij complex wordt. $\text{T}_\text{E}\text{X}$ biedt vele mogelijkheden tot het produceren van fraaie documenten. Het gebruik van de basiscommando's is, net als bij Troff, vrij ingewikkeld. Ook hier echter zijn macropakketten in ontwikkeling die de verwerking van documenten met een standaard

Invoertekst:

```
Na integratie verkrijgen we
$$ W(t) = 1 - \exp \{(-\int_0^t p(\alpha) d\alpha)\} $$
Stel  $p(t) = at$ . De voortgang wordt dan gegeven door een Rayleigh-verdeling:
$$ dw \frac{dt}{dt} = at \exp \{-at^2/2\} $$
```

Figuur 9: Voorbeeld van formules in T_EX.

layout sterk kunnen vereenvoudigen.

In [Knuth84] wordt een uitstekende beschrijving van T_EX gegeven. Tenslotte is er een gebruikersgroep die een nieuwsbrief verzorgt [TUGboat].

3.3. Scribe

Bij Troff, T_EX, en andere soortgelijke systemen bestaat de invoer uit een mengeling van tekst en commando's. De commando's zijn vaak van laag niveau en nemen de vorm aan van specifieke layoutaanwijzingen: "spring nu 3 spaties in", "ga over op font 7". De gebruiker is gedwongen zich met dit soort zaken bezig te houden wil hij precies dat op papier krijgen wat hem voor ogen staat. De opsteller van documenten heeft traditioneel deze kennis niet. Hij zou zich moeten beperken tot het leveren van de tekst en het aangeven van de (logische) structuur van het document. De verdere verwerking kan het beste geschieden door een expert in tekstopmaak en layout.

Een stap in de goede richting wordt gezet indien men bij systemen als Troff en T_EX gebruik maakt van macropakketten. In zo'n macropakket is dan de opmaak van een specifiek type document gedefinieerd en men maakt vervolgens gebruik van commando's als "begin een nieuwe paragraaf", "nu volgt een voetnoot", enz.

In Scribe is deze opzet veel verder doorgevoerd [Reid80]. Er wordt in Scribe gebruik gemaakt van een databank waarin voor verschillende *document typen*, zoals een zakenbrief, een intern memo, een artikel voor tijdschrift X, precies is vastgelegd hoe een tekst voorzien van aanwijzingen betreffende de structuur moet worden omgevormd tot een gezette versie. Het is de bedoeling dat wijzigingen en aanvullingen op deze databank alleen door experts worden aangebracht.

In figuur 10 is weergegeven hoe de uitvoer in figuur 3 via Scribe verkregen zou kunnen worden. Scribe stelt de gebruiker in staat op eenvoudige wijze de logische structuur van een document weer te geven, waarna alle layout-details verder door het systeem worden verzorgd. Anderzijds kunnen alleen die objecten verwerkt worden waarvoor beschrijvingen van de structuur en layout in de databank zijn opgenomen. Zo kunnen met de in [Reid80] beschreven versie geen complexe formules en tabellen verwerkt worden.

```
@Make (article)
@Heading (Software engineering)
@Center (J.C. van Vliet)
@PrefaceSection (Abstract)
In dit artikel wordt beargumenteerd dat Software Engineering als
verplicht vak op Universiteiten en Hogescholen ingevoerd zou
moeten worden.
@section (Inleiding)
Programmatuur wordt vaak te laat opgeleverd, voldoet niet aan
de specificaties, en bevat bovendien fouten.
@Subsection (Waarom Software Engineering ook een leuk vak is)
In tegenstelling tot Functietheorie is Software Engineering een vak
waarbij men ...
```

Figuur 10: Een invoertekst voor Scribe.

3.4. SGML

Binnen ISO vindt standaardisatie op het gebied van tekstverwerking plaats in TC97/SC18. Een van de belangrijke onderdelen van de in ontwikkeling zijnde standaard is SGML - "Standard Generalized Markup Language" [ISO85].

Ook in SGML ligt de nadruk sterk op het aangeven van de logische structuur van een document. Het systeem kan geïnformeerd worden over de precieze structuur en het type van elk tekstelement dat gebruikt mag worden. In een *document type definition* worden alle mogelijke correcte vormen van een bepaald type tekstelement opgegeven. In figuur 11 is een vereenvoudigde versie opgenomen van de declaratie van een element *fig*.

<!	ELEMENT	CONTENT	ATTRIBUTE
1	fig	(body, caption?)	id
2	body	(artwork (p ol) +)	
3	artwork	NONE	depth
4	caption	(#CDATA)	
>			

Figuur 11: SGML-declaratie van een 'figuur'.

In de eerste regel staat aangegeven dat een figuur een *body* bevat, mogelijk gevolgd door een *caption*. De "?" geeft aan dat de *caption* optioneel is. Een figuur heeft een attribute *id* waarin een identificatie gegeven kan worden. In de volgende regel staat gedefinieerd dat de *body* bestaat uit hetzij *artwork*, hetzij een reeks van paragrafen (*p*) en/of geordende lijsten (*ol*). De elementen *p* en *ol* moeten elders in de definitie van het document-type worden opgenomen. In het gebruikte formalisme leze men "of" voor "|", en "1 of meer maal herhaald" voor "+".

Vervolgens wordt *artwork* nader gedefinieerd. Er wordt een lege inhoud aangegeven - het plaatje wordt later ingeplakt -, en er is een attribuut *depth* waarin de hoogte kan worden gegeven. Tenslotte wordt gespecificeerd dat de *caption* bestaat uit een rij van 0 of meer karakters.

In essentie wordt hier een grammatica voor de structuur van het betreffende element gegeven in een BNF-achtige notatie.

Met deze formele definitie nu is de opmaak voor een plaatje als in figuur 12 toegestaan. De verschillende labels (tags) zijn tussen de symbolen "<" en ">" geplaatst. Elk onderdeel wordt begonnen met een beginlabel (als "<fig"). Een eindlabel neemt de vorm "<\fig>" aan. Op plaatsen waar dit niet tot ambiguiteiten aanleiding kan geven mag de eindlabel worden weggelaten.

```
<fig id=diagram1>
<body>
<artwork depth=20>
<caption> Een stroomdiagram
<\fig>
```

Figuur 12: Correcte opmaak voor een plaatje.

Men kan zo voor elk tekstelement een declaratie geven. Tezamen bieden deze een precieze beschrijving van een bepaald type document, en een gegeven tekst kan vervolgens aan de hand van deze beschrijving geanalyseerd worden. Indien de beschrijving niet toelaat dat er binnen een plaatje weer een plaatje voorkomt wordt een dergelijk voorkomen in een invoertekst gedetecteerd. Dit is analoog aan de manier waarop voor programmeertalen uitgaande van een grammatica een parser wordt gegenereerd met behulp van een parser-generator als Yacc [Johnson78] waarmee vervolgens programma's in de betreffende programmeertaal geanalyseerd worden.

Door de in SGML geboden faciliteiten voor het definiëren van de syntactische vorm van opmaakcommando's wordt aansluiting op reeds bestaande systemen mogelijk.

Het werk aan deze standaard is nog niet afgesloten. Voor zover de auteurs van dit artikel bekend, zijn er nog geen implementaties van dit systeem beschikbaar.

4. Interactieve verwerking van documenten

Implementaties van in de vorige sectie besproken systemen zijn batchgeoriënteerd: er wordt een invoerfile gecreëerd en zonder tussenkomst van de gebruiker wordt deze tot een opgemaakte tekst gevormd. In scherpe tegenstelling hiermee staan de interactieve tekstverwerkingssystemen, waar tekstopmaak- en grafische ontwerpfaciliteiten geïntegreerd zijn. Dat wil zeggen dat het voor de gebruiker mogelijk is binnen één zitting zowel platte tekst als grafieken en formules te maken, waarmee bovendien gemakkelijk gemanipuleerd kan worden. Een document is in deze visie niet meer een lineaire rij karakters, maar een object met een eigen structuur en subobjecten zoals tekst, plaatjes en formules.

Een tweede belangrijk aspect van interactieve tekstverwerking is het zogenaamde 'WYSIWYG' principe: What You See Is What You Get. Dit houdt in dat voor de gebruiker alle wijzigingen die hij in zijn document aanbrengt onmiddellijk zichtbaar moeten zijn. Dit is waar interactieve systemen essentieel verschillen van systemen uit de klasse van Troff en $T_E X$.

Ten derde willen we ingaan op wat met het modewoord 'gebruikersvriendelijkheid' wordt aangeduid. Men moet niet vergeten dat het belangrijkste afzetgebied voor fabrikanten van dit soort systemen het geautomatiseerde kantoor is, waaruit volgt dat een heterogene groep van gebruikers met relatief weinig moeite in staat moet zijn zich in een dergelijke omgeving te bewegen.

Voordat we overgaan tot het bespreken van een tweetal voorbeelden, lichten we kort de filosofie achter het ontwerp van een tegenwoordig snel populair wordende gebruikersomgeving toe. De voor dit soort omgevingen noodzakelijke hardware en software (grafische beeldschermen met hoge resolutie, muizen, ergonomisch vriendelijke toetsenborden en dergelijke) bestaat al geruime tijd, maar zijn pas de laatste jaren binnen het bereik van de financieel minder draagkrachtige consument gekomen.

Centraal staat de schrijftafel of 'desktop'. Voor de klassieke bureauwerker is dit een belangrijk deel van zijn beroepswereld: een schrijftafel met daarop een aantal (meer of minder ordelijk gerangschikte) documenten, een in- en uitbak voor post, etc. Een modern werkstation biedt in feite een beeld dat daar zoveel mogelijk op gemodelleerd is: objecten in het systeem (bestanden, programma's, e.d.) worden op het beeldscherm gerepresenteerd d.m.v. plaatjes, 'iconen' geheten. De gebruiker kan deze iconen manipuleren, waarbij men dan moet denken aan handelingen zoals openen, verplaatsen, versturen en opbergen in een archief. Dit manipuleren gebeurt met een zogenaamde muis: een doosje met een of meer drukknoppen dat over tafel bewogen kan worden. Hierbij rolt een zich binnen dit doosje bevindend balletje over het werkblad; deze bewegingen worden vertaald in bewegingen van een pijltje op het beeldscherm. Op deze manier worden iconen geselecteerd (door aanwijzen) en geactiveerd (door het bedienen van een of meer van de drukknoppen). Commando's die op eenmaal geselecteerde iconen werken worden gegeven in 'pop-up' menu's.

Terzijde moet hier opgemerkt worden dat er expliciet naar uniformiteit en consistentie van de commandostructuur gestreefd wordt: commando's krijgen in verschillende context toch zoveel mogelijk dezelfde betekenis.

De analogie met de schrijftafel wordt verder volgehouden door het 'window' begrip: de gebruiker krijgt uitzicht op een eenmaal geactiveerd object door een venster. Dit venster wordt als het ware over vorige vensters heen gelegd (die dan ook zichtbaar blijven voor zover ze niet door het nu actieve venster bedekt worden). Hierdoor wordt de indruk van een bureau bezaaid met papieren verder versterkt. Er zijn natuurlijk mogelijkheden om deze vensters onderling van plaats te doen verwisselen met de gebruikelijke selectieprincipes, ze te verplaatsen, van grootte te doen veranderen en dergelijke.

Het filesysteem tenslotte is georganiseerd als een ladenkast: files worden opgeborgen in laden, 'folders' genaamd. Dit filesysteem heeft veelal een boomvormige structuur: folders kunnen op hun beurt weer folders bevatten, etc.

We lichten het voorgaande toe met een korte bespreking van een tweetal op dit moment operationele systemen, de Xerox 8010 en het Edimath systeem.

4.1. De Xerox 8010

De Xerox 8010 is onderdeel van een uitgebreide familie van producten die samen veel behoeften dekken van het geautomatiseerde kantoor: de Xerox 8000 familie. Deze omvat o.a. word processors, fileservers voor gegevensopslag, een laserprinter en communicatiefaciliteiten, zowel lokaal (Ethernet interface) als globaal (opname van lokale Ethernet netwerken in een internationaal Internet netwerk). Behalve de normaal aanwezige onderdelen zoals een cpu, geheugen en dergelijke, heeft de Xerox 8010 een hoge resolutie (1024 x 809 pixels) bitmapped beeldscherm, een 29 Mbyte hard disk en een muis. Voor hardcopy- en communicatiefaciliteiten wordt een beroep gedaan op het Ethernet en een daarin opgenomen laserprinter (300 dots/inch).

Het gebruikersinterface is in grote lijnen als hierboven beschreven. Een document bijvoorbeeld wordt gecreëerd door een blanco document te activeren; op vergelijkbare wijze kan een bestaand document gewijzigd worden. Aan de gebruiker wordt in de teksteditor de controle geboden over tekstattributen zoals fontkeuze, fontgrootte, onderstreping, sub- en superscripts. Deze attributen zijn samengebracht in een 'property sheet'. De gebruiker kan een willekeurig stuk tekst selecteren en hiervan de attributen opvragen; deze verschijnen vervolgens in een apart venster, en hierin kunnen de gewenste veranderingen aangebracht worden. De nieuwe verzameling attributen kan dan van toepassing verklaard worden, hetgeen tot gevolg heeft dat de eerder geselecteerde tekst wordt aangepast.

Een soortgelijke editor wordt voor plaatjes geboden: complexe objecten kunnen opgebouwd worden uit basisvormen, waarbij er keuzemogelijkheden zijn voor o.a. vullen met een bepaald patroon, verschillende omlijning en vormverandering. Verder is er een grafiekgenerator die gedreven wordt door een door de gebruiker te verschaffen tabel. Hiermee kunnen pie- en barcharts gegenereerd worden. Op het scherm kunnen meerdere documenten tegelijk behandeld worden, waarbij er uitgebreide faciliteiten zijn voor het integreren van tekst- en grafische objecten.

De 8010 biedt tenslotte ook mogelijkheden tot het verwerken van formules. Hierbij maakt men o.a. gebruik van 'virtuele' toetsenborden: de gebruiker kan de toetsen op het toetsenbord laten corresponderen met bij voorbeeld Griekse letters, of tekens die in wiskundige formules worden gebruikt.

OPS-2000 is een met de 8010 vergelijkbaar systeem dat op een SUN werkstation draait [Markoff85]. Vele hierin opgenomen faciliteiten zijn terug te voeren tot onderzoek dat bij Xerox PARC is verricht (Smalltalk, Alto, Star).

4.2. Het Edimath systeem

Ons tweede voorbeeld is het bij het IMAG laboratorium o.l.v. V. Quint ontwikkelde Edimath systeem. Het betreft hier een omvangrijk researchproject dat nog slechts gedeeltelijk gerealiseerd is.

Bij dit systeem wordt uitgegaan van een abstracte visie op het onderwerp documentverwerking. Een document wordt gezien als een object met een subobjectstructuur en een formeel gedefinieerde syntax. Voorbeelden van objecten zijn tekst, plaatjes en formules. Een verdere doelstelling is dat het ontwerp onafhankelijk moet zijn van specifieke hardware.

Als prototype heeft men een editor voor wiskundige formules ontworpen, die inmiddels geïmplementeerd is op verschillende systemen. Voor twee van deze implementaties is moderne hardware zoals een hoge resolutie beeldscherm en een muis beschikbaar, voor de overige maakt men gebruik van standaard alfanumerieke terminals.

De formule editor is interactief en kent een uniforme commandostructuur: commando's worden gegeven door functietoetsen of menuselectie, en een bepaalde toets (menukeuze) heeft een vaste betekenis die tijds- en contextonafhankelijk is. Creëren en wijzigen van formules is volkomen syntaxgestuurd en men bevindt zich altijd in een welgedefinieerde constructie. De basisconstructies zijn: breuken, wortels, integralen, sommaties, vectoren, blokken, exponenten en karakterstrings. Bij het wijzigen van onderdelen van deze constructies worden de omringende constructies onmiddellijk meeveranderd; maakt men bijvoorbeeld de noemer van een breuk langer, dan wordt ook de breukstreep verlengd en worden teller en noemer opnieuw gecentreerd. De gehele formule is in een boomvormige structuur opgeslagen, en men kan zich (syntaxgestuurd) door deze boom verplaatsen.

F	$\frac{\bullet}{\bullet}$	E	$\frac{1}{2n} \int_0^{\sqrt{y}} \bullet$
1	$\frac{1\bullet}{\bullet}$	E	$\frac{1}{2n} \int_0^{\sqrt{y}} \bullet$
E	$\frac{1}{\bullet}$	B (	$\frac{1}{2n} \int_0^{\sqrt{y}} (\bullet$
2 n	$\frac{1}{2n\bullet}$	T H S	$\frac{1}{2n} \int_0^{\sqrt{y}} \left[\sum_{\bullet} \right]$
E	$\frac{1}{2n} \bullet$	k=1 E	$\frac{1}{2n} \int_0^{\sqrt{y}} \left[\sum_{k=1}^{\bullet} \right]$
I	$\frac{1}{2n} \int_{\bullet}$	n E	$\frac{1}{2n} \int_0^{\sqrt{y}} \left[\sum_{k=1}^n \bullet \right]$
0	$\frac{1}{2n} \int_{0\bullet}$	sin P	$\frac{1}{2n} \int_0^{\sqrt{y}} \left[\sum_{k=1}^n \sin^{\bullet} \right]$
E	$\frac{1}{2n} \int_0^{\bullet}$	etc...	
R	$\frac{1}{2n} \int_0^{\sqrt{\bullet}}$		
y	$\frac{1}{2n} \int_0^{\sqrt{y}\bullet}$		

Figuur 13: Voorbeeld van een sessie met Edimath.

Voor wijziging van constructies heeft men een vast aantal commando's tot zijn beschikking (delete/replace/insert before/insert after).

Bij wijze van voorbeeld is in figuur 13 een deel van een interactieve sessie met Edimath weergegeven. In de linkerkolom staan de door de gebruiker gegeven commando's, in de rechterkolom staat de formule zoals deze op het scherm getoond wordt. De vetgedrukte kakacters geven het gebruik van functietoetsen aan: Fraction, End, Integral, Root, Block, Power, enz. Het somteken wordt door een Triple gegeven. De H geeft aan dat het een Hoog symbool is, de S duidt op het Somteken. De plaats van de cursor wordt in de rechterkolom met het symbool $\bullet$ aangegeven.

Het systeem kent vele interessante implementatiedetails waarop we hier niet verder in kunnen gaan; zie ook [Quint83]. Vermeld zij slechts dat de layout van formules op Knuth's aanpak geënt is.

5. Conclusie

Het in sectie 4.2 beschreven Edimath systeem is een goed voorbeeld van hoe een systeem voor het verwerken van complex materiaal er uit zou kunnen zien. Sterk verbeterde gebruikersinterfaces vinden we o.a. bij de Xerox 8010, de Macintosh en het OPS-2000 systeem. De tijd is rijp voor systemen die verder gaan dan het op een beeldscherm imiteren van een schrijfmachine [Lemmons84].

Ook bij het Centrum voor Wiskunde en Informatica wordt onderzoek verricht op het gebied van interactieve tekstverwerking. Karakteristieken van het systeem dat ons voor ogen staat zijn:

- integratie van tekst, plaatjes, formules, e.d., in een systeem, waarbij de gebruiker een uniforme verzameling commando's tot zijn beschikking krijgt;
- de gebruiker ziet alleen de tekst zoals deze gezet zou worden, en de operaties sluiten op dit

beeld aan;

- het systeem is interactief, operaties hebben een onmiddellijk effect op hetgeen op het beeldscherm getoond wordt;
- de door de gebruiker te geven opdrachten hebben betrekking op de (logische) *structuur* van het document, en niet op de *layout*;
- de gebruiker heeft de mogelijkheid de uitlijnalgoritme te sturen (hij kan bij voorbeeld aangeven hoe slecht frequent afbreken van woorden is relatief ten opzichte van het toevoegen van extra wit);
- het systeem biedt eenvoudige mogelijkheden tot het corrigeren van teksten.

Wat dit laatste betreft kan men aan het volgende denken: bij het corrigeren van proeven maakt men veelal gebruik van een aantal tekens met elk een vaste betekenis, zoals bij voorbeeld vastgelegd in [NEN632]. Aldus gemerkte teksten zien er uit als in figuur 14 weergegeven. Men kan deze symbolen ook met een muis op een beeldscherm "tekenen". Onderzocht zou kunnen worden of dit tot een ergonomisch aantrekkelijke wijze van tekstcorrectie leidt.

Bij het met de hand corrigeren van drukproeven worden allerlei correctietekens gebruikt. Voorbeelden zijn tekens om woorden samen te voegen, letters weg te nemen, of aan te geven dat er geen overgang naar een nieuwe regel moet komen.

[Men kan aangeven dat er ingesprongen moet worden. Het systeem zou een dergelijke beweging van de muis kunnen herkennen en vertalen in het juiste commando voor de tekstverwerker.

Figuur 14: Het aangeven van correcties.

Ongetwijfeld zijn ook anderen in Nederland actief op het gebied van (interactieve) verwerking van documenten. De auteurs van dit artikel hebben het plan dit najaar een bijeenkomst te organiseren waarop over hiermee verband houdende vraagstukken van gedachten gewisseld kan worden. Geïnteresseerden in zo'n bijeenkomst wordt verzocht met een van ons contact op te nemen.

6. Referenties

- [Akkerhuis84] J. Akkerhuis, *Typesetting at the CWI*, CWI Newsletter 3, pp 15-24 (Deel I), CWI Newsletter 4, pp 21-32 (Deel II).
- [Barthel72] G. Barthel, *Konnte Adam schreiben? Weltgeschichte der Schrift*, M. DuMont Schauberg, Keulen, 1972.
- [Bigelow83] C. Bigelow & D. Day, *Digital typography*, Scientific American, August 1983, pp 94-112.
- [Bigelow85] C. Bigelow, *Font design for personal workstations*, BYTE, January 1985, pp 255-270.
- [Boot82] M. Boot & H. Koppelaar, *De tekstmachine*, Academic Service, Den Haag, 1982.
- [Brown84] H. Brown, *From text formatter to printer*, in [Miller84], pp 98-107.
- [Carter84] K.A. Carter, *IMP - a system for computer-aided typeface design*, in [Miller84], pp 114-119.
- [Diringer77] D. Diringer, *A history of the alphabet*, Unwin Brothers Ltd., Old Woking, 1977.
- [Flowers84] Jim Flowers, *Digital type manufacture: an interactive approach*, IEEE Computer 17, 5 (May 1984) pp 40-49.
- [Furuta82] R. Furuta, J. Scofield & A. Shaw, *Document formatting systems: survey, concepts, and issues*, ACM Computing Surveys 14, 3 (September 1982) pp 417-472.
- [Groenendaal75] M.H. Groenendaal, *Drukletters. Hun ontstaan en hun gebruik*, Technische Uitgeverij H. Stam N.V., Culemborg, 1975.
- [ISO85] Information-Processing Systems - Text Preparation and Interchange - Processing and Markup Languages - Part Six: *Generic Document Representation Specification (SGML)*, Draft Proposal International Standard ISO/DP8879/6, 1985.
- [Johnson78] S.C. Johnson, *Yacc: Yet Another Compiler Compiler*, Bell Laboratories, Murray Hill, NJ, 1978.
- [Kernighan75] B.W. Kernighan & L.L. Cherry, *A system for typesetting mathematics*, Communications of the ACM 18, 3 (maart 1975) pp 151-157.
- [Kernighan82a] B.W. Kernighan, *A typesetter-independent Troff*, Computing Science Technical Report 97, Bell Laboratories, Murray Hill, NJ, 1982.
- [Kernighan82b] B.W. Kernighan, *PIC - A language for typesetting graphics*, Software-Practice & Experience 12, 1 (januari 1982) pp 1-20.
- [Kernighan84] B.W. Kernighan, *The UNIX document preparation tools - a retrospective*, in [Miller84] pp 12-25.

- [Knuth79] D.E. Knuth, *T_EX and METAFONT, new directions in typesetting*, Digital Press, Bedford, MA, 1979.
- [Knuth81] D.E. Knuth & M.F. Plass, *Breaking paragraphs into lines*, *Software-Practice & Experience* 11, 11 (november 1981) pp 1119-1184.
- [Knuth84] D.E. Knuth, *The T_EXbook*, Addison-Wesley Publ. Cy., 1984.
- [Lemmons84] Ph. Lemmons, *Beyond the word processor*, *BYTE* (januari 1984) pp 53-56.
- [Lesk78] M.E. Lesk, *Typing documents on the UNIX system: using the -ms macros with Troff and Nroff*, Bell Laboratories, Murray Hill, NJ, 1978.
- [Lesk79] M.E. Lesk, *TBL - A program to format tables*, Bell Laboratories, Murray Hill, NJ, 1979.
- [Markoff85] J. Markoff & Ph. Robinson, *Lasers, office publishing, and more*, *BYTE* (april 1985) pp 379-384.
- [Meyrowitz82] N. Meyrowitz & A. van Dam, *Interactive editing systems, Part I, II*, *ACM Computing Surveys* 14, 3 (september 1982) pp 321-416.
- [Miller84] J.J.H. Miller (Ed.), *PROTEXT I, Proceedings of the First International Conference on Text Processing Systems*, Boole Press Ltd., Dublin, 1984.
- [NEN632] *Correctietekens voor drukproeven*, Nederlands Normalisatie-Instituut, NEN 632.
- [Nievergelt82] J. Nievergelt, G. Coray, J.-D. Nicoud & A.C. Shaw, *Document preparation systems*, North-Holland Publishing Cy, Amsterdam, 1982.
- [Ossanna76] J.F. Ossanna, *Nroff/Troff user's manual*, Bell Laboratories, Murray Hill, NJ, 1976.
- [Quint83] V. Quint, *An interactive system for mathematical text processing*, *Technology and Science of Informatics* 2, 3 (1983) pp 169-179.
- [Reid80] B.K. Reid, *Scribe: a document specification language and its compiler*, PhD Thesis, Dept. of Computer Science, Carnegie-Mellon University, 1980.
- [Sigplan81] *Proceedings of the ACM SIGPLAN SIGOA Symposium on Text Manipulation*, *ACM SIGPLAN Notices* 16, 6 (June 1981).
- [Spivak83] M. Spivak, *Summary of A_MS-T_EX*, *TUGboat* 4, 2 (september 1983) pp 103-126.
- [Treebus83] K.F. Treebus, *Tekstwijzer, een gids voor het grafisch verwerken van tekst*, Staatsuitgeverij, Den Haag, 1983.
- [TUGboat] Newsletter of the T_EX Users Group, American Mathematical Society, PO Box 6248, Providence, RI 02940.
- [Van Vliet72] J.C. van Vliet, *Bestesplits1, een procedure voor het automatisch afbreken van Nederlandse woorden*, rapport IN28/72, Centrum voor Wiskunde en Informatica, 1972.