

# Computerondersteund tekstgebruik

**J. Hagen**

Onderwijskundig Bureau voor Advies- en Ontwikkelwerk,  
Postbus 125,  
8000 AC Zwolle

11 januari 1994

## Context

Naast opleidingsmateriaal en rapporten, produceert PRAGMA teksten die veelal een functie als naslagwerk vervullen. Hierbij kan men denken aan praktische studieteksten, werkplekinstructies en handleidingen bij machines. De keuze voor  $\text{\TeX}$  is destijds ingegeven door de hoge typografische kwaliteit, flexibiliteit en onafhankelijkheid van dit systeem.

In toenemende mate wordt informatie aangeboden via een beeldscherm, bijvoorbeeld handboeken en handleidingen. Vaak laat zowel de typografische kwaliteit als de kwaliteit van de weergave op het scherm te wensen over. Tegenover het nadeel van een matig leescomfort staat het voordeel van betere toegankelijkheid en relatief eenvoudige onderhoudbaarheid. De media papier en beeldscherm vullen elkaar dan ook eerder aan dan dat zij elkaar uitsluiten.

## Aanleiding

Onlangs hebben wij onze dvi-processoren ingeruild voor DVIPSONE en DVIWINDO, beiden van Y&Y. Wij zijn over deze programma's zeer tevreden, zij het dat DVIWINDO figuren in een lage resolutie toont (preview). In dit programma kunnen met behulp van het `\special`-commando koppelingen tussen tekstdelen worden gelegd. Dat wil zeggen dat het aanklikken van een bepaald tekstdeel resulteert in het gaan naar een ander tekstdeel. Hiermee komt een vorm van computerondersteund tekstgebruik binnen bereik, waarbij een hoge typografische kwaliteit wordt gehandhaafd. Dit was voor ons reden om de mogelijkheden van computerondersteund tekstgebruik nader te onderzoeken.

## Eisen

Binnen PRAGMA maken we gebruik van een eigen macropakket. Vanzelfsprekend maakt een referentiemechanisme deel uit van dit pakket. Het is mogelijk te verwijzen naar alle daarvoor in aanmerking komende elementen van een tekst, zoals hoofdstukken, paragrafen, figuren, tabellen, formules, definitives, opsommingen, voetnoten enz.

Er is meer nodig dan een referentiemechanisme om een tekst op de computer toegankelijk te maken. Ook het aanklikken van onderdelen in overzichten, zoals een inhoudsopgave, en ingangen in indices moet resulteren in

het naar een bepaalde positie gaan. Bij voorkeur moet op elke bladzijde naar relevante overzichten kunnen worden gesprongen, bijvoorbeeld vanuit een menu. Ook moeten woorden in de tekst kunnen worden aangeklikt. Tenslotte ligt een koppeling tussen (delen van) figuren en tekst voor de hand. Samengevat dient een interactiemechanisme dus tenminste de volgende functionaliteit te bezitten:

1. springen naar het onderwerp van een verwijzing
2. springen naar een onderdeel van een lijst of een index
3. springen naar uitleg bij een woord
4. springen naar uitleg van (delen van) figuren
5. springen vanuit een menu naar lijsten, hoofdstukken enz.

Eis 1 brengt met zich mee dat de plaats van de verwijzing wordt gemarkeerd. Klikken op *zie figuur 1* moet resulteren in een sprong naar de betreffende figuur.

Voldoen aan eis 2 maakt het nodig de onderdelen van lijsten en ingangen in een index te markeren, zowel in de lijst als in de tekst zelf. Bij een index moet klikken op een paginanummer resulteren in een sprong naar de relevante pagina én plaats in de tekst. De voorkeur gaat uit naar klikken op een paginanummer, omdat bij een ingang meerdere verwijzingen mogelijk zijn. In een inhoudsopgave moet het klikken op een hoofdstuknummer of de tekst leiden tot een sprong naar het betreffende hoofdstuk. Hetzelfde geldt voor overzichten van figuren, tabellen enz. Alle hiervoor noodzakelijke verwijzingen moeten door  $\text{\TeX}$  (macro's) automatisch worden gegenereerd.

Eis 3 is een variant op een standaard referentiemechanisme en heeft alleen typografische gevolgen: visueel moet duidelijk zijn welke woorden aan te klikken zijn.

Om in een figuur een onderdeel te kunnen aanwijzen (eis 4), moet een deel van het figuur kunnen worden gedefinieerd als 'button'. Het daarvoor gebruikte mechanisme moet nauw samenwerken met zowel de plaatsing van de figuur in de tekst als het schalen van de figuur. Verwijzen moet twee kanten op kunnen: vanuit de tekst naar bijvoorbeeld een onderdeel van een afgebeelde machine of vanuit de tekst over bijvoorbeeld een kenmerk van een gebouw naar een representatief element op een foto. Dit alles kan worden verwezenlijkt door over de figuur een (meegehaalde) onzichtbare matrix te leggen.

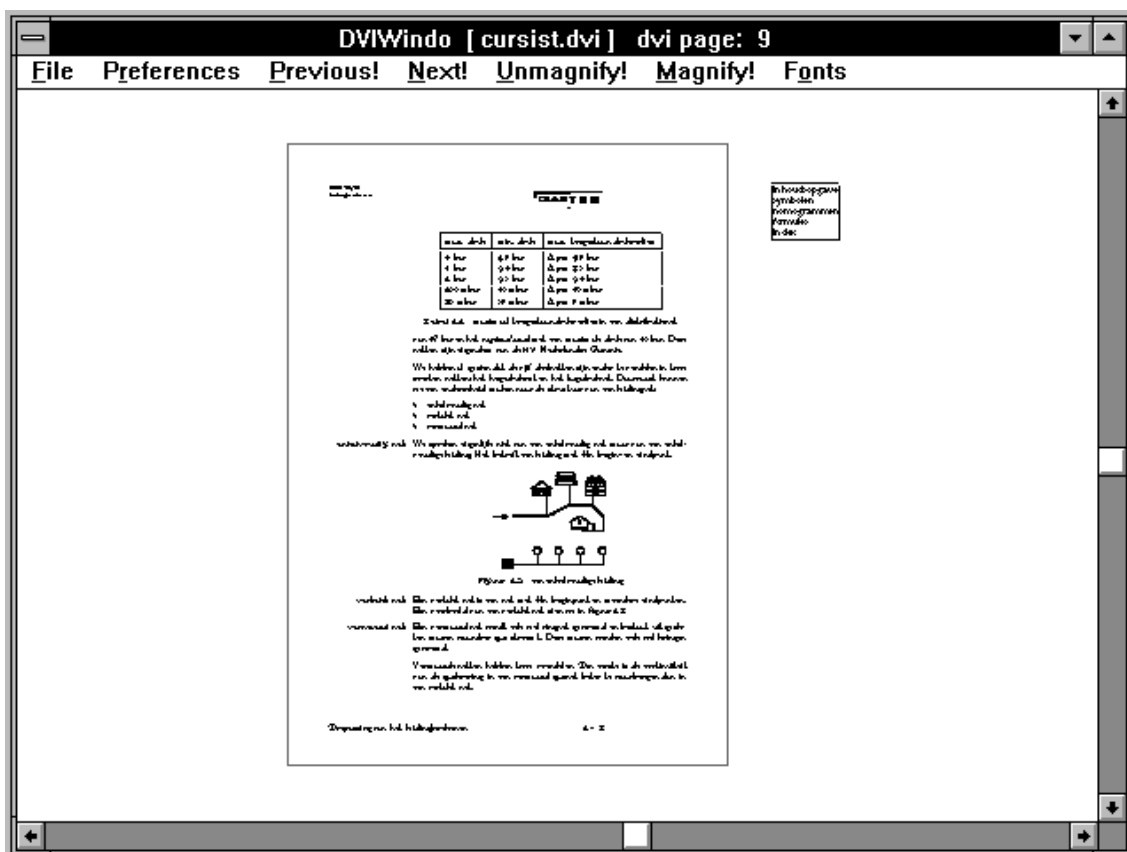
De laatste eis tenslotte, heeft tot gevolg dat in de output-routine aan elke bladzijde een menu wordt toegevoegd met eerder gedefinieerde keuze-items. Deze items kunnen weer door middel van het gebruikelijke referentiemechanisme aan hoofdstukken, paragrafen, bladzijden enz. worden gekoppeld. Hierop aansluitend moeten de voorvieren gebruikte programma's wel de mogelijkheid bieden naar de vorige plaats in de tekst terug te gaan.

Als extra randvoorwaarde kan nog worden genoemd, dat in een tekst zo min mogelijk extra zetcommando's moeten worden toegevoegd. Een tekst moet bij voorkeur zowel

geschikt zijn voor weergave op papier als voor raadplegen op de computer.

## Implementatie

De genoemde functionaliteit is binnen PRAGMA inmiddels volledig gerealiseerd. Daarbij worden twee (pre)viewers ondersteund. Het eerder genoemde DVIWINDO stelt de minste eisen, omdat de koppelingen door middel van labels tot stand komen. Het is in dit programma mogelijk de cursor exact op de gevraagde plaats te positioneren. De oplossing is ook relatief goedkoop: de previewer kost slechts enkele honderden gulden.



Figuur 1: Een voorbeeld van de weergave van tekst in dviwindo

In figuur 1 is een voorbeeld van een scherm weergegeven. Rechts van de tekst is een in  $\text{\TeX}$  aangemaakt menu zichtbaar. Aanklikken van een item in dit menu resulteert in een sprong naar de inhoudsopgave, de index, een overzicht van formules enz.

Uitgaande van het beschikbaar zijn van een volwassen referentiemechanisme, kunnen in DVIWINDO relatief eenvoudig teksten met een, weliswaar beperkte interactiviteit worden geproduceerd. Dit gebeurt met behulp van de commando's:

```
\special{mark: "label"}
```

en

```
\special{button: width height "label"}
```

Met het eerste commando markeren we een plaats in de tekst en met het tweede commando maken we een (onzichtbare) 'button' aan. Klikte men met de muis op deze button, dan wordt naar de plaats met hetzelfde label gesprongen.

Sinds kort brengt ADOBE het programma ACROBAT op de markt. Met dit programma kunnen PostScript-files worden omgezet in het pdf-formaat (portable display format). Files in dit formaat kunnen op verschillende platforms worden bekeken. Door middel van het  $\backslash$ special-commando kunnen pdf-commando's in de PostScript-file worden opgenomen:

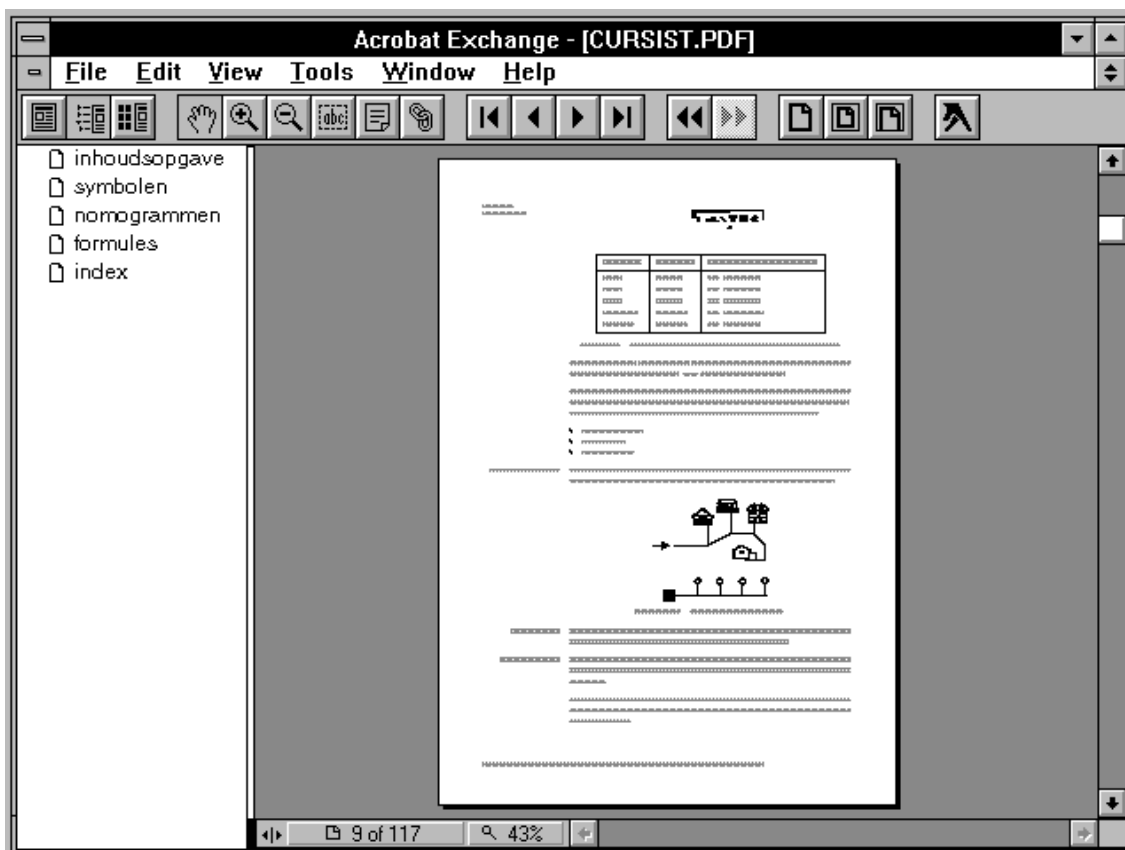
```
\special{postscript [/ ... .. pdf]}
```

Helaas kan alleen naar (delen van) pagina's worden gesprongen. Dit betekent dat bij elke verwijzing naast het

gezette paginanummer het volgnummer moet worden bewaard. Er wordt dus niet gesprongen naar bladzijde 3–5 of VIII maar naar volgnummer 23. Bovendien springt de cursor niet naar de gevraagde plaats in de tekst of lokatie in een figuur. Dit is vanuit  $\text{\TeX}$  niet eenvoudig in waterdichte pdf-aanroepen te vertalen. Het verbaast mij trouwens dat het enkele duizenden guldens kostende ACROBAT niet tenminste het label-mechanisme van DVIWINDO biedt. Overigens gaat men er kennelijk vanuit dat de gebruikers van

het pakket zelf geen software ontwikkelen. De bijgeleverde documentatie is minimaal en de foutmeldingen vallen buiten het niet te vergroten window.

De eerder genoemde menu's kunnen in ACROBAT worden geïmplementeerd met behulp van 'bookmarks'. Deze leveren links op het scherm een verkorte inhoudsopgave op. In figuur 2 is een tekst weergegeven waarbij rechts een menu zichtbaar is met dezelfde keuzemogelijkheden als in figuur 1.



Figuur 2: Een voorbeeld van de weergave van tekst in acrobat

De bovenstaande referentie-mechanismen vreten geheugen, met name uit  $\text{\TeX}$ 's stringpool. Een 'big'  $\text{\TeX}$ -implementatie, gecombineerd met macro's die doelmatig gebruik maken van de stringpool, is dan ook noodzakelijk, omdat interactieve documenten meestal omvangrijk zijn. In  $\text{em}\text{\TeX}$  kan trouwens de stringpool meer dan voldoende worden vergroot.

Samengevat zijn er dus twee mechanismen naast elkaar nodig om teksten in meerdere (pre)viewpakketten interactief te kunnen bekijken: 'link by label', zoals in DVIWINDO, en 'link by page', zoals in ACROBAT.

## Ontwikkelingen

Waar DVIWINDO uitblinkt in eenvoud en snelheid, biedt ACROBAT op termijn meer mogelijkheden. Het is bijvoorbeeld voor lezers mogelijk aantekeningen te maken in de

elektronische tekst. Daarnaast worden figuren in een hoge resolutie weergegeven.

Het zou mooi zijn als in de nabije toekomst in ACROBAT ook bewegende beelden en geluidsfragmenten kunnen worden opgeroepen. In plaats van terug te moeten vallen op voor wat betreft zetwerk matige pakketten, kan dan  $\text{\TeX}$  worden gebruikt als basis, met alle inmiddels bekende voordelen van dien.

Een viewer, uitgaande van dvi-files zou, naast het besproken label-mechanisme, ook de mogelijkheid moeten bieden via  $\backslash\text{special}$ 's delen van de tekst te verbergen, dit om geneste menu's te kunnen implementeren. Ook zou een soort `execute program` commando moeten worden ondersteund. Externe programma's kunnen dan bijvoorbeeld bewegende beelden laten zien. Dergelijke mogelijkheden worden op dit moment bij PRAGMA verder onderzocht en ontwikkeld.