

# Een moderne toren van Babel

**Johannes L. Braams**

## Samenvatting

In deze voordracht zal ik kort aandacht besteden aan ‘wat is  $\text{T}_{\text{E}}\text{X}$ ’, ‘wat is  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ’. Hierbij komt ook het concept van ‘structuur markup’ aan de orde. Vervolgens zal ik ingaan op de geschiedenis van het **babel** pakket. Aan de hand van een aantal voorbeelden zal ik toelichten welke problematiek zich voordoet bij het op een correcte wijze vormgeven van documenten waarin meerdere talen worden gebruikt.

---

Een moderne toren van Babel

Overzicht

### Overzicht

1. Inleiding
2. Wat is  $\text{T}_{\text{E}}\text{X}$  en  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
3. Wat is structuur markup?
4. De geschiedenis van babel
5. Meertalige documenten

---

21 November 1996

$\text{T}_{\text{E}}\text{X}$ niek

2

## 1 Inleiding

Wanneer een auteur een document moet maken staan hem daartoe in principe twee verschillende *typen* hulpmiddelen ter beschikking.

---

Talen en Tools

Inleiding

### Inleiding

Tekstproductie:

WYSIWIG of  
markup?

---

21 November 1996

$\text{T}_{\text{E}}\text{X}$ niek

3

Het meest gebruikte type hulpmiddel is de WYSIWIG<sup>1</sup> tekstverwerker. Voorbeelden hiervan zijn ‘Word for Windows’ en Wordperfect for Windows’. Deze hulpmiddelen vereisen een grafische user interface en zijn derhalve voornamelijk voor personal computers<sup>2</sup> beschikbaar.

In de Unix wereld zijn hulpmiddelen van het tweede type, de ‘markup taal’ beter beschikbaar. Voorbeelden hiervan zijn troff en  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .

Tussen de gebruikers van beide typen ‘tekstverwerkers’ woedt soms net zo’n heftige ‘geloofs’-strijd als tussen de gebruikers van VMS en Unix. Een voordeel van het eerste type hulpmiddel is dat de gebruiker directe terugkoppeling krijgt over hoe zijn document er op papier ongeveer uit zal gaan zien. Een voordeel van het tweede hulpmiddel is dat de auteur zich op enig moment (tot op zekere hoogte) alleen met de inhoud (en niet, of in ieder geval minder, met de vorm) van zijn document bezig houdt. De vorm is dus voor een belangrijk deel losgekoppeld van de inhoud.

## 2 Wat is $\text{T}_{\text{E}}\text{X}$ en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ?

$\text{T}_{\text{E}}\text{X}$  is een programma voor het ‘zetten’ van teksten, ontwikkeld door professor Donald E. Knuth van de Stanford Universiteit. De auteur heeft  $\text{T}_{\text{E}}\text{X}$  ontwikkeld omdat de drukkerij die zijn serie boeken ‘The Art of Computer Programming’ drukte met andere apparatuur ging werken waardoor zijn boeken een ander aanzien kregen. Hij heeft het als een uitdaging gezien om met behulp van de computer minstens dezelfde typografische kwaliteit te bereiken die voorheen met door een zetter de zetmachine werd geleverd. Omdat ook het lettertype waarin zijn boeken gezet werden niet meer beschikbaar was heeft hij ook het programma METAFONT ontwikkeld waarmee hij het lettertype ‘Monotype Modern 8a’ als ‘computer modern roman’ opnieuw het licht deed zien.

---

<sup>1</sup>WYSIWIG staat voor ‘What you see is what you get’. In de praktijk blijkt dit vaak net niet helemaal op te gaan. . .

<sup>2</sup>in de breedste betekenis; een Apple Macintosh valt voor mij ook onder de ‘personal computers’.

Wat is T<sub>E</sub>X?

T<sub>E</sub>X is een programma om  
met behulp van de computer  
hoge kwaliteit zetwerk voor  
*The Art of Computer Programming*  
te produceren.

Professor Knuth ontwikkelde een programma, T<sub>E</sub>X (spreek uit: ‘tech’) genaamd, dat als invoer de tekst van een document krijgt, vermengd met instructies die het programma duidelijk maken hoe de tekst op papier moet worden gezet. Teneinde het gebruik van T<sub>E</sub>X zo flexibel mogelijk te laten zijn heeft de auteur T<sub>E</sub>X voorzien van een macro taal. Dat wil zeggen: het programma T<sub>E</sub>X zelf kent een beperkt aantal instructies (ongeveer 300 zogenaamde primitives). De instructieset is eenvoudig uit te breiden door nieuwe instructies te definiëren op basis van bestaande instructies. Professor Knuth heeft met het beschikbaar stellen van T<sub>E</sub>X ook een wat uitgebreidere instructieset (ook wel macropakket genoemd) ter beschikking gesteld, ook wel bekend onder de naam PLAIN T<sub>E</sub>X. Later zijn hierop door verschillende andere auteurs uitbreidingen op gemaakt. De bekendste, en met name in Europa, meest verspreide uitbreiding is het door Leslie Lamport ontwikkelde L<sup>A</sup>T<sub>E</sub>X.

Wat is L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X is een uitbreiding op  
de instructieset van T<sub>E</sub>X.

L<sup>A</sup>T<sub>E</sub>X kent een taal die  
de *structuur* van  
een document beschrijft.

Eén van de belangrijkste kenmerken van L<sup>A</sup>T<sub>E</sub>X is dat de ‘taal’ van L<sup>A</sup>T<sub>E</sub>X op een ander niveau is gedefinieerd. Vergeleken met programmeertalen is de verhouding tussen PLAIN T<sub>E</sub>X en L<sup>A</sup>T<sub>E</sub>X overeenkomstig die tussen assembler en PASCAL.

### 3 Wat is structuur markup?

De term ‘markup’ wordt gebruikt voor commando’s die tussen de normale tekst worden geplaatst en die als instructie gelden voor diegene die ervoor moet zorgen dat de tekst vermenigvuldigd kan gaan worden. In het Nederlands wordt ook wel de term ‘zetinstructies’ gebruikt,

hoewel die term naar mijn idee voornamelijk voor de één na laatste stap in het productieproces van toepassing is. Zetinstructies schrijven bijvoorbeeld voor dat een hoofdstuk in een boek op een rechter pagina moet beginnen; dat de hoofdstuktitel een bepaalde afstand naar beneden op de pagina gezet moet worden uit een specifiek corps, etc. Dergelijke markup wordt ook wel visuele markup genoemd.

## Wat is structuur markup?

- Visuele markup
 

```
‘start een rechter pagina’;
‘skip 25 mm witruimte’;
‘zet het hoofdstuknummer uit Arial corps
18’
2
‘skip 15 mm witruimte’;
‘zet de hoofdstuktitel uit Arial corps
20’
De titel van het hoofdstuk
‘skip 15 mm witruimte’
```

Wanneer de commando’s die tussen de tekst zijn gevoegd niet meer direct de te bereiken visuele effecten op de pagina beschrijven, maar in hoofdzaak iets zeggen over de ‘logische elementen’ van een document spreken we over ‘structuur markup’. Een dergelijke taal bevat onder meer commando’s waarmee aangegeven kan worden dat een nieuw hoofdstuk begint en wat de titel is van dat hoofdstuk, waar een genummerde lijst begint en eindigt, waar de afzonderlijke elementen van die lijst beginnen etc. Een document dat op een dergelijke wijze van markup is voorzien is dan ook veel beter geschikt om hergebruikt te worden in bijvoorbeeld database toepassingen. Doorgaans wordt onafhankelijk van het document dat de markup bevat, bepaald hoe de gebruikte commando’s leiden tot bepaalde visuele effecten op papier (of op een beeldscherm).

## Wat is structuur markup?

- Structuur markup
 

```
‘doctype:artikel’
‘taal:nederlands’
‘chapter:De titel van het hoofdstuk’
```

Bovenstaand voorbeelden illustreren naar mijn idee een aantal belangrijke verschillen tussen visuele markup en structuur markup. In het tweede geval is het 100% zeker dat *elke* hoofdstuktitel er consequent hetzelfde uitziet. In het eerste geval kan makkelijk een fout gemaakt worden wat uiteindelijk een slordige indruk op de lezer zal maken.

In L<sup>A</sup>T<sub>E</sub>X zijn dergelijke ideeën over ‘structuur markup’ in belangrijke mate terug te vinden.

## 4 De geschiedenis van babel

Tegen de tijd dat de populariteit van het programma T<sub>E</sub>X zodanig groot was dat ook mensen in Europa ermee aan de slag gingen had Leslie Lamport L<sup>A</sup>T<sub>E</sub>X al uitgevonden. Het gevolg was dat in Europa relatief meer gebruik gemaakt werd (en wordt) van L<sup>A</sup>T<sub>E</sub>X dan van PLAIN T<sub>E</sub>X. Maar L<sup>A</sup>T<sub>E</sub>X is een van oorsprong Amerikaans produkt. Dus een hoofdstuk begon altijd, hard gecodeerd, met het woord ‘Chapter’; *ook* als het om een nederlandstalig document ging. Hiervoor werden in Europa op verschillende plaatsen door verschillende mensen gelijksoortige oplossingen voor bedacht. Met name in Duitsland en Frankrijk waren aanpassingen van L<sup>A</sup>T<sub>E</sub>X in omloop die aan dit soort problemen het hoofd trachtten te bieden. In die tijd was ook in Nederland een schare enthousiaste gebruikers van L<sup>A</sup>T<sub>E</sub>X ontstaan die zich opnieuw met dit probleem geconfronteerd zagen. De eerste ‘oplossing’ was om de uitbreidingen die voor het Duits in Duitsland al waren gerealiseerd voor het Nederlands over te nemen (voor zover relevant). Hierin ligt de oorsprong van het babel pakket.

Talen en Tools Het ontstaan van Babel

### Het ontstaan van Babel

- Verschillende, gelijksoortige, *nationale* uitbreidingen voor de ondersteuning van de eigen taal door L<sup>A</sup>T<sub>E</sub>X;
- Veel gezamenlijke programmatuur;
- *Babel*: Breng alle gezamenlijke programmatuur onder in één gemeenschappelijke ‘kernel’ en laat de ondersteuning voor de verschillende talen daar gebruik van maken.

21 November 1996 T<sub>E</sub>Xniek 8

Op de EuroT<sub>E</sub>X conferentie in Karlsruhe (1989) begonnen mijn ideeën meer vorm te krijgen, met name naar aanleiding van discussies met een aantal conferentie deelnemers. In de eerste helft van 1990 kwam de eerste release van *babel* beschikbaar. Op dat moment was er ondersteuning voor een half dozijn talen. Niet lang na het verschijnen van *babel* begonnen mensen bijdragen te leveren om het aantal ondersteunde talen te laten groeien. Momenteel worden 31 verschillende talen ondersteund.

Talen en Tools Ontwikkeling van Babel

### Ontwikkeling van Babel

- Eerste versie in 1990; 6 talen worden ondersteund.
- Tweede versie in 1992; meer talen, ‘stack’ van macro definities ten behoeve van ‘genest’ talen.
- Derde grote aanpassing 1994–1995; nog meer talen; introductie van het begrip ‘shorthand’.

21 November 1996 T<sub>E</sub>Xniek 9

De ondersteuning voor sommige van die talen gaat niet verder dan het ter beschikking stellen van de juiste vertalingen voor woorden als ‘chapter’, ‘figure’ etc. Voor andere talen gaat de ondersteuning veel verder; soms wordt de invoer van letters met diacritische tekens vereenvoudigd, soms worden speciale typografische gewoonten ondersteund.

Talen en Tools Hoofdfuncties van Babel

### Hoofdfuncties van Babel

- Kiezen van de juiste afbreekregels;
- Kiezen van de juiste vertalingen van standaardwoorden;
- Kiezen van de juiste vertaling en vorm van de datum;
- Inschakelen van taal-specifieke conventies;

21 November 1996 T<sub>E</sub>Xniek 10

Talen en Tools taal-specifieke conventies

### taal-specifieke conventies

- Vereenvoudiging van invoer,
- Speciale afbreekregels (zoals verdubbeling van letters of verdwijnen van een trema bij afbreken),
- bijzondere effecten.

21 November 1996 T<sub>E</sub>Xniek 11

De belangrijkste functie van *babel* is het selecteren van het juiste afbreek algoritme. Zoals in figuur 1 te geïllustreerd wordt is dat tamelijk belangrijk. (Kijk er de dagbladen maar op na!) In dezelfde figuur wordt ook de tweede functie van *babel* getoond, namelijk de vertaling van het woord ‘figure’ in het rechter deel van de figuur. Eén van de taal-specifieke conventies voor het Nederlands tenslotte, is de ondersteuning van het lage aanhalingsteken openen (,,).

## 5 Meertalige documenten

Er kunnen twee klassen van meertalige documenten onderscheiden worden. Ten eerste zijn er documenten waarin een ‘hoofdtal’ wordt gebruikt, maar waarin ook passages uit één of meer andere talen voorkomen. Voor dergeli-

## Voorbeeld

De  
geïrri-  
teerde  
mede-  
wer-  
ker  
ge-  
bruikte  
een  
tek-  
stver-  
werker.

Figure 1: Nederlandse tekst, engels afgebroken

De  
geïrri-  
teerde  
mede-  
wer-  
ker  
ge-  
bruikte  
een  
tekst-  
ver-  
wer-  
ker.

Figuur 2: Nederlandse tekst, nederlands afgebroken

Figuur 1: Vergelijking tussen het afbreken van nederlandse tekst met een engels en een nederlands afbreek algoritme

jke documenten zijn de vertalingen van de standaardwoorden minder van belang, maar de auteur kan er aan hechten wél zoveel mogelijk de juiste typografische conventies te hanteren voor elke taal. Tenminste moet gezorgd worden dat woorden zoveel mogelijk correct worden afgebroken.

Het programmaboekje van deze conferentie kan als voorbeeld dienen van een dergelijk document. Het bevat enkele abstracts in het Engels, terwijl de het grootste deel van de tekst in het Nederlands is gesteld.

Een tweede klasse van documenten is die waarin twee of meer talen een min of meer gelijkwaardige rol spelen. Voorbeelden van dergelijke documenten kunnen met name gevonden worden bij de EU. Sommige publicaties bevatten dezelfde inhoud meermaals, in verschillende talen. Andere gebruiken voor elk hoofdstuk een andere officiële taal.

## Meertalige documenten

1. Eén 'hoofdtaal' met passages uit andere talen
2. Meerdere gelijkwaardige 'hoofdtalen'