
Nederlandstalige T_EX Gebruikersgroep

MAPS: M_inutes and AP_pendiceS#16 (96.1)

Verslag:

1.	Opening 16 ^e NTG bijeenkomst 5 september 1995	1
2.	Verslag NTG bijeenkomst van 24 mei 1995	1
3.	Ingekomen stukken en mededelingen	1
4.	Begroting 1996	1
5.	Verslag TUG'95	1
6.	Rondvraag en sluiting	2
7.	Volgende bijeenkomsten	2

Bijlagen:

A	Het weten waard	3
B	Van de Voorzitter	4
C	Van uw MAPS Editor	5
D	Financieel verslag 1995	7
E	Jaarverslag NTG	9
F	Van de TUG President	12
G	Bericht van FGBBS	14
H	A plug-and-play teTeX CD-ROM	16
I	Attending EuroTeX'95 in Papendal	17
J	BachoTeX 1996	24
K	The 17th Annual TeX Users Group Meeting	26
L	De kunst van het programmeren	30
M	An Interview with Donald Knuth	33
N	Knuth meets NTG members	38
O	TeX without 4TeX on the PC	50
P	Van dictaat tot boek	54
Q	Enkele ervaringen met L ^A T _E X	57
R	Een L ^A T _E X-cursus in Groningen	58
S	Het gebruik van L ^A T _E X voor wiskunde lessen	59
T	Van ms-troff naar L ^A T _E X	60
U	The L ^A T _E X2HTML translator: An Overview	61
V	Zijn Scientific Word en TeX uitwisselbaar?	74
W	Typografisch programmeren	76
X	Paradigms: It's all in the game	82
Y	Paradigms: The winds and halfwinds—Details matter	85
Z	BLUe-2-L ^A T _E X—expansion and some more	91
A	Graphics and TeX—a reappraisal of METAFONT/Metapost	100
B	Page layout in L ^A T _E X	108
C	Colored Verbatim	126
D	NTG ledenlijst 1996	140

Nederlandstalige T_EX Gebruikersgroep (NTG)

Voorzitter:	E.H.M. Frambach RUG, Econometrie, Groningen E-mail: e.h.m.frambach@eco.rug.nl	
Secretaris:	G.J.H. van Nes ECN, Unit Faciliteiten, Petten E-mail: vannes@ecn.nl	
Penningmeester:	W. Dol LEI-DLO, Den Haag E-mail: w.dol@lei.dlo.nl	
Bestuursleden:	J.L. Braams E-mail: jlbraams@cistron.nl	F. Goddijn E-mail: goddijn@fgbbs.iaf.nl
Postadres:	Nederlandstalige T _E X Gebruikersgroep Postbus 394 1740 AJ Schagen	
Postgiro:	1306238 t.n.v. Penningmeester NTG Eindhoven	000-1662209-17 t.n.v. Ph. Vanoverbeke (NTG) Langenhoekstraat 28, B-8210 Veldegem, België
E-mail bestuur:	ntg@nic.surfnet.nl	

T_EX is een, door professor Donald E. Knuth ontwikkelde, ‘opmaaktaal’ voor het letterzetten van documenten, een documentopmaakstelsel. Met T_EX is het mogelijk om kwalitatief hoogstaand drukwerk te vervaardigen. Het is eveneens zeer geschikt voor formules in wiskundige teksten.

Er is een aantal op T_EX gebaseerde producten, waarmee ook de logische structuur van een document beschreven kan worden, met behoud van de letterzetmogelijkheden van T_EX. Voorbeelden zijn L^AT_EX van Leslie Lamport en $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_EX van Michael Spivak.

De **Nederlandstalige T_EX Gebruikersgroep (NTG)** is een vereniging die tot doel heeft het bevorderen van de kennis en het gebruik van T_EX.

De NTG tracht dat te bereiken door het uitwisselen van informatie, het organiseren van congressen en symposia m.b.t. T_EX en ‘T_EX-producten’ en door het onderzoeken en vergelijken van T_EX met soortgelijke/aanverwante producten.

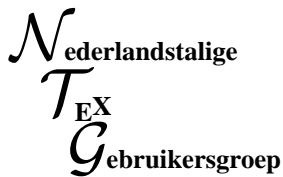
De NTG biedt haar leden onder meer het volgende:

- Tweemaal per jaar een NTG-bijeenkomst.
- Tweemaal per jaar het uitgebreide NTG tijdschrift: MAPS (Minutes and Appendices).
- Speciale MAPS uitgaven (o.a. T_EX-cursus, FAQ, en PR-set).
- De *dubbele* 4allT_EX CD-ROM met een volledige en direct te gebruiken T_EX DOS/Windows/OS-2/Linux implementatie inclusief een uitgebreide verzameling van utilities. De CD-ROM bevat zeer veel documentatie, inclusief discussielijsten van vele jaren, *alle* MAPS uitgaven, en zeer veel tutorials.
- Korting op (buitenlandse) T_EX congressen en cursussen en op het lidmaatschap van TUG.
- Jaarlijks een ledenlijst met informatie over welke software/hardware, in relatie met T_EX, wordt gebruikt.
- De discussielijst (listserver) TEX-NL waarop vragen worden gesteld en ervaringen uitgewisseld.
- De fileservers TEX-NL waarop algemeen te gebruiken ‘T_EX-producten’ staan.
- Het NTG FGBBS Bulletin Board met ruim 800 MByte aan T_EX en aanverwante software.
- Activiteiten in werkgroepen. Enkele belangrijke werkgroepen zijn: ‘Nederlandse T_EX’, ‘PC’s en T_EX’, ‘educatie’ (cursussen), en ‘communicatie’.

Lid worden kan door overmaking aan de penningmeester van het verschuldigde contributiebedrag. Daarnaast dient een informatieformulier te worden ingevuld, dat via het secretariaat te verkrijgen is.

De contributie voor een persoonlijk lidmaatschap bedraagt f 90,-, de contributie voor een instituutslidmaatschap bedraagt f 245,-. Een instituutslidmaatschap geeft het recht om drie personen aan te wijzen die informatie welke aan de leden wordt verstuurd, ontvangen. Van die drie personen dient één persoon te worden aangewezen als rechtsgeldige vertegenwoordiger van het bedrijf/instituut, een ander als vervangend vertegenwoordiger. Indien meer leden per bedrijf/instituut lid willen worden, geldt als additioneel tarief f 65,- per persoon. Voor studenten geldt een tarief van f 60,- (geen stemrecht; bewijs van inschrijving vereist). Voor afwijkende regelingen dient contact met het bestuur opgenomen te worden. Een gecombineerd NTG/TUG lidmaatschap voor 1994 bedraagt f 167,50 per jaar (i.p.v. f 90,- + \$ 55).

Belgische leden kunnen de lidmaatschapskosten van BF 1660 (individueel), BF 4520 (instituutslidmaatschap) of BF 3090 (NTG/TUG lidmaatschap) overmaken op de NTG Belgische postgiro te Veldegem (zie hierboven).



- Aanwezig** : A.W.W.M. Biegstraaten (TUD); P. Bloemen; J. Braams (T_EXniek); L. de Coninck (de Kraal); N. Cox (KUN); G. van den Dobbelsteen (LOGIN BV); W. Dol; E. Frambach (RUG); G. Geens; F. Goddijn (FGBBS); M. Goossens (CERN); J. Hagen (Pragma); R. van der Heijden (Hogeschool van Utrecht); T. Hoekwater (Wolters Kluwer Academic Publishers); N.S. Kroonenberg (RUG); C.G. van der Laan; F.D. Mesman (Elsevier Science Publishers); G.J.H. van Nes (ECN); N. Ramdjan; G. Oomen (Wolters Kluwer Academic Publishers P. van Oostrum (UU); S.A.M. Pepping (Elsevier Science Publishers); F. Poppe (SWOV); J. Pijnenburg (KUB); J. Renkema (TU); R. Rietman (Philips Natuurkundig Laboratorium); E. de Rouville; A.C.A. Serier; J.H. van de Stadt; P. Tutelaers (TUE); G. Vlak; J. van Weelden; J.E. van Weerden (UU);
- Notulist** : Frans Goddijn

1 Opening

Voorzitter Erik Frambach opent de vijftiende NTG-bijeenkomst.

2 Verslag van de NTG-bijeenkomst van 24 mei 1995

'Piet van Oostrom' (blad 1, kolom 2) moet zijn 'Piet van Oostrum'.

Bij punt 5, 'Verslag commissie voor kascontrole' staat Jules van Weerden ten onrechte genoemd. De commissie bestaat uit Phons Bloemen en Simon Pepping.

Met deze correcties worden de notulen goedgekeurd.

3 Ingekomen stukken en mededelingen

Gerard van Nes noemt verschillende binnengekomen periodes van onze zusterverenigingen. Van het GUTenberg Cahier wordt via TEX-NL een inhoudsopgave verspreid, waarna belangstellenden een artikel toegezonden kunnen krijgen.

Kees van der Laan meldt zich met de mededeling dat hij de rest van de vergadering op pad moet gaan met de onder zijn hoede gebrachte T_EX-ers uit het oosten.

4 Begroting 1996

Wietse Dol vertelt dat het financieel beheer van het NTG een stuk gecompliceerder is geworden, terwijl een behoorlijk aantal leden traag is van betalen. Nu de contributie pas in de loop van het jaar binnenkomt, is het moeilijker om een eventueel TUG-lidmaatschap goed te laten sporen. Immers, de latere betaling zorgt voor een (te) late aanmelding

bij TUG met alle gevolgen van dien.

We zoeken nog naar een oplossing voor sommige 'instuutsliden' die een factuur nodig hebben voor hun contributie.

De MAPS is voortaan aan een vast maximum pagina's gebonden.

Bij punt 9 van de begroting voor 1996 dient in plaats van '1995' te worden gelezen: '1996'.

De vergadering gaat akkoord met de begroting.

5 Verslag TUG'95

Wietse Dol spreekt uit dat de gelegde contacten er hij hem toe hebben geleid dat zijn vertrouwen in de mogelijkheden van TUG is hersteld.

Michel Goossens (niet slechts TUG-president, maar ook NTG-lid) hecht zeer aan reacties en suggesties van TUG-leden. Hij pleit er vurig voor dat TUG-leden niet weglopen maar erbij blijven om samen de zaak weer op te bouwen. Michel legt uit welke oorzaken hebben geleid tot het niet-verschijnen van een aantal TUGboats. Dat blijkt voornamelijk te hebben gelegen aan de historisch gegroeide exclusieve bemoeienis van Barbara Beeton met het blad, haar onbekendheid met L^AT_EX₂ε en reorganisaties op haar werkplek. Nu wordt er een systeem opgezet waarbinnen elke schakel is te vervangen door iemand anders.

Michel beoogt TUG waarlijk internationaal te maken, zodat elk LUG-lid ook TUG-lid zal zijn. De oude situatie van 5 'special directors' is achterhaald en ook door de toelieferanciers van deze special directors terzijde geschoven. Een nieuwe opzet waarbinnen de LUGs van buiten de USA hun invloed kunnen laten gelden is nog niet gevonden.

Gerrit Oomen komt spontaan met de suggestie dat TUG zich zou organiseren op de wijze zoals de Bridge Bond zich via gewesten en districten heeft georganiseerd.

6 Rondvraag en sluiting

- Wietse Dol maakt bekend dat hij overweegt te stoppen als penningmeester, vooral uit teleurstelling voortkomend uit het tegenvallende betalingsgedrag van de leden en ook door de tegenvallende inschrijving van NTG-leden voor Euro \TeX .
- Erik Frambach hamert er als voorzitter op dat er meer vrijwilligers nodig zijn om o.a. te helpen bij de administratieve taken.
- De volgende vrijwilligers melden zich:
 - Phons Bloemen: administratie & MAPS
 - Gilbert van den Dobbelensteen: MAPS
 - Taco Hoekwater: algemeen
 - Siep Kroonenberg: MAPS en vormgeving
- Erik Frambach vertelt dat de volgende bijeenkomst zal plaatsvinden op zaterdag 25 mei 1996 (zou nog een week verschuiven) in de Hogeschool van Amsterdam met als thema 'L \AA TeX in MBO en HBO'.
- Erik Frambach meldt dat Johannes Braams werk zal maken van een inventarisatie van actieve, slapende en in stilte opgeheven werkgroepen.
- Phons Bloemen meldt dat de WWW-werkgroep van hem, Jules van Weerden en Henk de Haan goed draait. De WWW-server is 'in de lucht'. De 'domain' aanvraag moet wegens een storing opnieuw gebeuren.
- Erik Frambach vraagt de aanwezigen om er eens over na te denken wat de NTG over een jaar of wat zou kunnen doen ter gelegenheid van het tweede lustrum.
- Michel Goossens wijst erop dat van 28 juli tot en met 1 augustus de TUG bijeenkomst in Rusland zal plaatsvinden. De omgeving is bijzonder geschikt om er een familie-uitstapje van te maken.
- Gerard van Nes vertelt dat naast de 4all \TeX CD-ROM van het NTG er een vraag begint te ontstaan naar een UNIX-variant. Michel Goossens maakt bekend dat er binnen TUG wordt gekeken naar de mogelijkheid om binnen 2 tot 3 maanden met zo'n CD-ROM voor UNIX te komen. Men zoekt samenwerking met anderen hiertoe. Sebastian Rahtz is de contactpersoon.
- Jules van Weerden wijst erop dat op onze NTG 4all \TeX CD-ROM ook GhostScript staat, en dit, gecombineerd met het werken op een UNIX systeem, zou security-problemen kunnen geven, aangezien PostScript de mogelijkheid in zich heeft om systeemopdrachten uit te voeren.

Erik Frambach sluit de vergadering en de deelnemers zijn de rest van de dag te gast bij Euro \TeX '95.

7 Volgende bijeenkomsten

De voorlopige planning van de volgende bijeenkomsten ziet er als volgt uit:

- **voorjaar 1996**
Hogeschool van Amsterdam: zaterdag 1 juni 1995
Thema: '(L \AA)TeX in het Middelbaar- en Hoger Onderwijs'
- **najaar 1996**
Faculteit Letteren, Universiteit Utrecht: donderdag 24 oktober 1996
- **voorjaar 1997**
Technische Universiteit Delft: mei 1997

1 \TeX kalender 1993/1994

1 jun '96	NTG (17°)	Amsterdam
28 jul 2 aug '96	TUG '96	Dubna, Rusland
najaar '96	NTG (18°)	Utrecht

2 Glossary

Gebruikersgroepen

TUG	:	\TeX Users Group
LUG	:	Local Users Group
CSTUG	:	LUG Tsjecho Slowakije
CyrTUG	:	LUG USSR
DANTE	:	LUG Duitsland
GUTenberg	:	LUG Frankrijk
HunTUG	:	LUG Hongarije
ITALIC	:	LUG Ierland
JTUG	:	LUG Japan
Nordic	:	LUG Scandinavië, Denemarken, en IJsland
NTG	:	LUG Nederland en België
SibTUG	:	LUG Siberië
UKTUG	:	LUG Engeland
YUNUS	:	LUG Turkije
GUST	:	LUG Polen

Bulletins/journals

Baskerville	:	UKTUG
Cahiers GUTenberg	:	GUTenberg
Zpravodaj	:	CSTUG
TeXnische Komödie	:	DANTE
TeXline	:	Malcolm Clark; UK
GUST bulletin	:	GUST
TTN	:	\TeX and TUG News; TUG
TUGboat	:	TUG
MAPS	:	Minutes and APpendiceS; NTG

Diversen

AMS	:	American Mathematical Society
BoD	:	Board of Directors
SGML	:	Standard Generalized Markup Language
ltxiii	:	\LaTeX 3.0
FGBBS	:	NTG's Bulletin Board

3 NTG's \TeX Bulletin Board System

Op het \TeX Bulletin Board van de Nederlandstalige \TeX Gebruikersgroep (FGBBS) is een zo volledig en actueel mogelijke \TeX , em \TeX , \LaTeX , TEX-NL en Music \TeX collectie beschikbaar voor alle bezitters van een modem. Het BBS is kosteloos toegankelijk voor iedereen en er zijn geen beperkingen aan de hoeveelheid bestanden die kunnen worden opgevraagd. Het systeem is aangesloten op een High Speed modem, vergeleken met de transmissiesnelheid die een directe Internet link biedt misschien niet

geweldig, maar veel beter kan het niet over de gewone huis- tuin- en keuken PTTlijn. De beheerder is Frans Goddijn. FGBBS is te bellen op 026 321 70 41.

4 NTG's winkel

Via de NTG is beschikbaar:

- **4all \TeX CD-ROM:**

Ruim 1 Giga byte aan ondermeer \LaTeX utilities, fonts, \TeX / \LaTeX /METAFONT/etc documentatie, de volledige MAPS 1 t/m 14, discussielijsten TEX-NL, TEX-HAX, UKTEX van de afgelopen 5 jaren, etcetc. Kosten inclusief het uitgebreide \LaTeX manual: f 60,-.

- **Syllabus Advanced \TeX course:**

Insights and Hindsights, David Salomon (*revised*; ruim 500 pagina's). MAPS'92 speciale uitgave. Kosten f 50,- voor leden en f 60,- voor niet-leden (extra verzendkosten: f 10,-).

- **PR set MAPS'93 speciale uitgave:**

Ruim 25 pagina's; 1 exemplaar gratis voor leden; extra exemplaren: f 2.50; niet-leden: f 5,- (extra verzendkosten: f 5,-).

Bestellingen kunnen gedaan worden door overmaking van het verschuldigd bedrag (plus verzendkosten) op de postgiro van NTG (1306238) t.n.v. penningmeester NTG, Eindhoven, met vermelding van hetgeen gewenst is.

5 MAPS 96.2

Sluitingsdatum voor het inleveren van artikelen, bijlagen, en/of mededelingen voor de volgende MAPS uitgaven is:

1 oktober '96 (MAPS 96.2) en 1 april '97 (MAPS 97.1).

Aanleveren kopij voor de komende MAPS:

- *Bij voorkeur* in \LaTeX gebruikmakend van de: `maps.sty` of `maps.cls` Deze stijlfile is via de redactie te verkrijgen en beschikbaar op de TEX-NL fileservers, archive.cs.ruu.nl (ftp-site) en FGBBS (026 321 70 41).
Daarnaast kunnen bijdragen ingestuurd worden gemaakt met `ltugboat` of `article/report style/class files`.
- Verder zijn bijdragen vanzelfsprekend ook welkom in *plain- \TeX* of ongeformatteerd.
- Plaatjes bij voorkeur als (Encapsulated) PostScript file.

Bij onduidelijkheid gaarne contact opnemen met de redactie.

Daar MAPS bijdragen in plain \TeX worden omgezet naar \LaTeX , verdient vanzelfsprekend aanbieding van materiaal in \LaTeX de voorkeur!

Eventuele nadere richtlijnen voor auteurs zijn op te vragen bij de redactie.

Bijdrage kunnen gestuurd worden naar:

`w.dol@lei.dlo.nl`

Niet Internet-gebruikers kunnen hun bijdrage ook via modem/PTT lijn direct naar de redactie sturen. Gaarne hiervoor eerst contact opnemen met Wietse Dol.

Van de Voorzitter

Erik Frambach

Laatst las ik in PC Magazine een anecdote die aardig genoeg is om ook in de T_EX-wereld eens bij stil te staan.

De anecdote gaat over een stel dat naar New York gaat en daar onder de indruk raakt van al die hoge gebouwen en het vele verkeer dat af en aan rijdt. Overal worden gebouwen opgericht en op andere plaatsen weer afgebroken. De man zegt verbijsterd tegen zijn vrouw: “Misschien kunnen we beter een keer terug komen wanneer alles af is.”

Vertaald naar de computerwereld, inclusief de T_EX-wereld, betekent het dat voordat een produkt volledig uitontwikkeld is er alweer een opvolger komt. We schamperen zo graag over produkten als WordPerfect die op dit moment met een stuk of drie niet-compatibele versies op de markt is. De gebruikers die van verschillende bronnen bestanden krijgen aangeleverd schieten volledig in de stress. En wij, verstandige T_EX-gebruikers, lachen minzaam om zoveel domheid. Wij kennen immers het genot van een stabiel systeem dat al vele jaren niet meer verandert en over 10 jaar nog zo zal zijn.

Maar zo simpel ligt het niet. Ook in de T_EX-wereld zien we steeds meer dat programma's snel vervangen worden door nieuwe, kennelijk nog betere, van nog meer snuffjes voorziene opvolgers. ‘Featuritis’ wordt dat wel genoemd. Wie weet nog precies wat de nieuwste versie van DVIPS is, van L^AT_EX, Babel, emT_EX, L^AT_EX2HTML en noem maar op?

Het is dan ook geen wonder dat de 4allT_EX cd-rom zo goed verkoopt, want niemand heeft zin, laat staan tijd, om van alles de nieuwste versies bij elkaar te zoeken én de zaak zo op te zetten dat alles met elkaar wil samenwerken.

Ik vraag me dan ook serieus af in hoeverre *gebruikers* vragen om al die nieuwe snuffjes. Zeker, er zijn mensen die dat graag zien en dat zijn ook de mensen die erom vragen. Maar hoe zit het met de overgrote meerderheid die je *niet* hoort? Die lijdt onder de druk om toch maar bij te blijven want als straks iets niet werkt is geheid het advies om de nieuwste versie te installeren. En wie heeft er al eens een nieuw pakket geïnstalleerd dat *kleiner* was de het voorgaande en *sneller*? Extra functionaliteit is mooi, maar nooit ten koste van basisfunctionaliteit.

Het lijkt erop dat Knuth nog steeds een roepende in de woestijn is. Zijn systeem is bevroren, het is af, en het mag gezien en bewonderd worden. Wanneer zou L^AT_EX af zijn, om maar eens een voorbeeld te noemen waar erg veel mensen van afhankelijk zijn. Bij veel pakketten krijg ik de indruk dat ze niet eens ernaar *streven* ooit af te zullen zijn. De wijsheid ‘stilstand is achteruitgang’ is tegelijk een halve leugen.

Genoeg gekankerd. Gelukkig zijn er ook lichtpuntjes te melden. Op het gebied van standaardisatie wordt op het moment druk gewerkt. Er is een commissie bezig een standaard te ontwikkelen voor DVI-drivers. Als dat lukt wordt het een stuk eenvoudiger om met plaatjes om te gaan, zonder je te hoeven verdiepen in de de syntaxis van `\specials` die bij elke driver weer anders is. Dat past in T_EX's filosofie: een DVI behoort device independant te zijn, dus ook DVI-driver independant.

Het lijkt er nu op dat de TDS, de T_EX Directory Structure, door gaat breken. De multi-Unix cd-rom die binnenkort uitkomt zal geheel volgens deze nieuwe standaard zijn opgebouwd. Nu nog afwachten of dat in de praktijk bevalt en wat het voor de performance betekent. De volgende versie van de 4allT_EX cd-rom zal ook de TDS aanhouden. Toch moeten we niet verwachten dat nu iedereen over zal stappen. Zoals ik eerder al heb aangegeven zijn er goede redenen om een goed werkend systeem niet overhoop te gooien. Zo komt Knuth zelf nog regelmatig versies van zijn fonts tegen die al vele jaren verouderd zijn, hetgeen zich verraadt door de minder geslaagde vorm van het delta-teken.

Een andere trend (in de gunstige zin van het woord) die opvalt is het terugdringen van bitmaps ten gunste van vectorgeoriënteerde systemen en, heel opmerkelijk, een herwaardering van ASCII-teksten. T_EX en METAFONT blijken in staat zich aan moderne eisen aan te passen. Sterker nog, T_EX kan heel goed functioneren als basis van waaruit o.a. HTML, PostScript en PDF wordt gegenereerd. T_EX is daarmee de spin midden in het (World Wide) web, en kan een tweede jeugd beginnen. Concurrenten zijn er volop, maar eentje met zo veel souplesse vind je nergens. Hooguit kan SMGL een betere basis zijn, waarbij T_EX nog steeds een rol kan spelen als *front end* voor het zetwerk.

Van oorsprong is T_EX, of eigenlijk meer METAFONT bitmap-georiënteerd. Het gebruik van PostScript Type 1 fonts is echter al heel gewoon. Het pakket MFTOEPS (BOP, Polen) heeft al zijn steentje bijgedragen, maar Metapost (John Hobby) zorgt voor een echte doorbraak. Knuth zelf is een fanatiek gebruiker. De bitmaps zijn daarmee bijna geëlimineerd. Wat nog ontbreekt is een simpele methode om fonts in METAFONT-code in Type 1 om te zetten. De Rus Malishev heeft daar programmatuur voor ontwikkeld en hopelijk wordt die spoedig beschikbaar gesteld.

T_EX en METAFONT zijn duidelijk springlevend en kunnen, mits op de juiste manier toegepast, een leven lang meegaan. Lees eerst nauwkeurig Knuth's gebruiksaanwijzing.

Van uw MAPS Editor

Gerard van Nes / Wietse Dol

Een nieuwe MAPS redactie

Het is een tijd van komen en een tijd van gaan. Na een periode van 15 MAPS-uitgaven (ruim 2100 pagina's) moet het maar eens tijd zijn de eindverantwoordelijkheid van de MAPS aan iemand anders over te dragen. Wietse Dol heeft met het maken van de proceedings van de Euro \TeX '95 conferentie duidelijk laten zien dat hij in staat is om in 2 weken tijd het gehele proces van 'benaderen/herbenaderen van auteurs' tot en met 'het aanbieden van de uiteindelijke PostScript files aan de drukker', volledig te beheersen. Najaar 1995 was daarom een goed tijdstip om de MAPS eindverantwoordelijkheid uit handen van Gerard van Nes te halen en over te dragen aan Wietse Dol.

Wietse kon al bij MAPS #11 verwelkomt worden. Aanvankelijk voor het oplossen van de pittigste problemen doch uiteindelijk voor het samenstellen van een gehele proceedings. Frisse wind zal nu door hem geblazen worden. Veel succes wordt hem hierbij toegewenst.

Voor Gerard zijn superlatieven niet genoeg. Een ieder die de produktie van een MAPS eens heeft meegemaakt weet hoeveel werk het is. Niet alleen de hoeveelheid maar ook de korte tijdsbestek waarin alles moet gebeuren geeft veel stress. We zijn als NTG dan ook hem dankbaar voor alle tijd, energie en moeite. Rest ons alleen nog hem zijn verdiende rust te gunnen.

De redactie is dit keer uitgebreid met Gilbert Dobbelsesteen. Gilbert is meteen in het diepe gegooid en heeft reeds aangetoond stressbestendig te zijn. Dit suggereert veel goeds voor de komende MAPS'en. Echter er kunnen nooit genoeg editors zijn. Dus als u zich geroepen voelt, laat het ons weten.

Natuurlijk zitten er in elke redactie stille krachten. Jos Winnink is er één. Zoals gebruikelijk heeft hij weer zonder veel gekreun zijn portie aan de MAPS bijgedragen.

Dit keer is zeer veel werk verricht door Christina Thiele. Onze Canadese editor wist in weinig tijd de 3 uur die Knuth aan vragen had beantwoord in leesbare tekst om te zetten. Voorwaar geen eenvoudige taak. Hulde en dank aan Christina!

Voor de nieuwe redactie blijft de steun gelden van zowel schrijvers als aandragers van interessant materiaal. Vergeet niet dat de NTG steunt op de kracht van allen. (L \AA) \TeX , PS, PDF, LP, EP, WEB, SGML, HTML, etc bijdragen zijn altijd welkom. Een half A4-tje met wat eigen (of andermans...) leuke ideeën is zo gemaakt. Een bijdrage van 10 à 20 bladzijden van elke NTG-lid is echter veel te veel!

Acht jaar MAPS

De MAPS-werkzaamheden zijn de laatste 8 jaar duidelijk aan verandering onderhevig geweest. De eerste MAPS in 1988 van 30 bladzijden, werd bijvoorbeeld bijeen gehouden met een eenvoudig ringbandje en verscheen in een op-

lage van ongeveer 100 stuks (de NTG kende toen 90 officiële leden). Het was toen wel een *echte* MAPS: Minutes met slechts enkele Appendices. Pas bij MAPS #5 werden de 156 bladzijden 'gebonden' in de bekende beige kaft. En dat bleef een kleine tien MAPS-uitgaven doorgaan. De oplage groeide naar ruim 300 stuks en het aantal bladzijden per MAPS varieerde tussen de 130 en 242 (MAPS #11: de 'jubileum-MAPS').

De MAPS-layout was altijd compact (veel informatie per pagina), 'eenvoudig' en zeer stabiel. De kwaliteit groeide van 300dpi kopieerkwaliteit (vanaf een matig tot slecht origineel) tot 600dpi DocuTech output. MAPS #14 bevatte dankzij een aantal donateurs zelfs enkele kleurenpagina's! De verwerking van artikelen evolueerde ook duidelijk. Een VAX-750 maalde langzaam de eerste MAPS-uitgaven erdoor. Een SUN-3 zorgde vervolgens voor een duidelijke snellere doorlooptijd, en tegenwoordige is het maken van een MAPS op een Pentium met (natuurlijk) de 4all \TeX CD-ROM een 'peuleschilletje'.

Wat gelukkig echter weinige veranderde was de *bereidwilligheid van vele auteurs* uit zowel binnen- als buitenland. Zij hebben in de afgelopen jaren meegeholpen om de MAPS op een goede en veelzijdige manier te vullen. Veelzijdigheid was ook het uitgangspunt van de MAPS: bijdragen zowel bestemd voor de doorgewinterde \TeX -guru als de net beginnende L \AA \TeX -gebruiker.

De oude MAPS-uitgaven bevatten materiaal wat nog steeds zeer actueel is. Slechts weinigen zullen in het bezit zijn van *alle* 15 gedrukte uitgaven. Echter sinds het vorig jaar integraal aanmaken van zowel de PostScript- als PDF-bestanden van *alle oude MAPS-uitgaven* is het materiaal nu ook volledig beschikbaar voor alle 4all \TeX CD-ROM bezitters! Zo kan men zelfs nog het verslag van de eerste NTG-bijeenkomst lezen...

De MAPS #16 bijdragen

Deze MAPS is wat dunner dan de voorgaande doch zeker niet minder interessant! U vindt onder meer een drietal unieke bijdrage m.b.t. Donald E. Knuth. Deze \TeX -grootheid was enkele dagen in Amsterdam en de NTG liet natuurlijk niet de gelegenheid voorbij gaan om met succes een '*ad hoc* bijeenkomst' te organiseren (weer één van die goede suggesties van ere-lid Kees van der Laan en meteen voortvarend aangepakt door Erik Frambach). Een vol zaaltje, goede vragen en dito antwoorden, levendige discussie's én een elektronische registratie t.b.v. diegenen die deze avond niet aanwezig konden zijn (vanwege de korte organisatietijd was namelijk alleen een aankondiging naar TEX-NL gestuurd). Een *letterlijke* uiteenzetting van de avond vindt u daarom ook in deze MAPS dankzij de gigantische medewerking van onze Canadese MAPS-editor Christina Thiele.

Verder weer veel bijdragen ‘voor elk wat wils’, waaronder nieuws van het conversieprogramma LATEX2HTML. Dit programma is een uitkomst voor elke WWW-pagina bouwer (voorbeelden van L^AT_EX documenten die via dit programma zijn geconverteerd zijn bijvoorbeeld te vinden op <http://www.ecn.nl/>, rubriek ‘documenten’).

Zowel L^AT_EX als *plain/non-plain*-T_EX gebruikers komen ook weer volop aan hun trekken. Natuurlijk zijn de auteurs Piet van Oostrum en Kees van der Laan weer present met zeer interessant materiaal over respectievelijk de nieuwe FANCYHEADINGS en diverse T_EX-vernuftigheden. Hans Hagen kan men ondertussen wel ook tot de vaste groep van auteurs rekenen met diverse *al dan niet gekleurde culinaire hoogstandjes* uit de PRAGMA-keuken. Ook opgenomen, vele zeer leesbare bijdragen op het gebied van L^AT_EX in

het onderwijs, een gloedheet verslag van BachoT_EX (10 dagen voordat de MAPS op uw deurmat lag eindigde deze Poolse bijeenkomst), informatie over de komende UNIX CD-ROM en nog veel meer.

De productie van de MAPS zit er bij de redactieleden weer op. Wij hopen dat deze uitgave u weer vele nieuwe ideeën en veel leesplezier zal opleveren.

U heeft het wellicht al gezien: de layout van de voorkant is veranderd. De nieuwe voorkant is gemaakt door Siep Kroonenberg. Dit als eerbetoon aan Gerard. Deze verandering is niet de laatste. De komende MAPS zal wellicht volledig nieuw gelayout worden. Dus u kunt de komende MAPSen nog iets verwachten.

Financieel verslag 1995

Nederlandstalige T_EX Gebruikersgroep

Wietse Dol

Januari 1996

1 Inleiding

Voor U ligt het financieel jaarverslag van de vereniging *NTG* over het jaar 1995. De financiële ontwikkeling van de vereniging wordt aan de hand van de winst en verliesrekening en de balans besproken. Om een goede indruk te krijgen van het verschil tussen de begroting en realisatie van 1995 is tevens in tabel 1 de begroting van 1995 opgenomen.

2 De winst en verliesrekening

In tabel 2 is de winst- en verliesrekening over 1995 weergegeven. De diverse posten worden in de volgende paragrafen kort toegelicht.

<i>Bedragen in guldens</i>	Uitgaven	Inkomsten
Contributie		21.500,00
Sponsoring	2.000,00	
Bijdrage Bursary fund	1.000,00	
Rente		1.400,00
Bijeenkomsten		
Administratie	800,00	
Kamer van Koophandel	61,00	
Bestuurskosten	400,00	
Computerfaciliteiten	PM	
Nieuwsbrief/Verslagen	16.200,00	
Reis bijdragen	2.000,00	
Representatie	200,00	
Afschrijving	1.182,00	
Onvoorzien	57,00	
Begroot verlies		1.000,00
	23.900,00	23.900,00

Tabel 1: *Begroting voor 1995*

2.1 Inkomsten

- **Contributies** Met bijna 300 leden is de NTG wederom gegroeid. De begroting ging van een minder aantal leden uit. Er staat nog zo'n f1500,00 aan contributies open. Meer leden betekent meer inkomsten maar ook direct meer uitgaven (vooral MAPSen).
- **Rente** De rente is aanzienlijk lager dan begroot. Dit komt omdat het rente percentage laag is en tevens omdat het geld lange tijd niet op de giro rekening heeft gestaan, maar is benut voor het aanschaffen van een voorraad CD-roms.

- **Evenementen** Geld wat is uitgegeven en ontvangen aan NTG-dagen en bezoeken aan instituten. Voor euroT_EX is een aparte rekening geopend en valt derhalve volledig buiten de NTG boekhouding. Wel zijn er diverse proceedings verkocht, welke gedeeltelijk tot opbrengsten van de NTG zijn gerekend.
- **Verkoop MAPS specials** Diverse oude en nieuwe MAPS specials verkocht voor de somma van f345,00.

<i>Bedragen in guldens</i>	Debet	Credit
Contributie		25.492,91
Sponsoring	1.000,00	
Bijdrage Bursary fund	1.000,00	
Rente		869,25
Evenementen		149,30
Administratie	502,10	
KvK en notaris	82,00	
Bestuurskosten	923,00	
Computer fac.		
Mededelingen a/d leden	19.645,61	
Reisbijdragen		
Representatie		345,00
Verkoop MAPS specials		
Verkoop CD-rom 1994	14.187,88	
Afschrijving apparatuur	1.182,05	
Diversen		
Saldo		11.666,18
	38.522,64	38.522,64

Tabel 2: *De winst en verliesrekening over 1995*

2.2 Uitgaven

- **Sponsoring** Dit jaar is f1000,00 gegeven aan het FGBBS. Het L^AT_EX-3 team had een dusdanige geld voorraad dat besloten is dit jaar geen sponsoring te verlenen.
- **Bursary fund** Er is f1.000,00 besteed aan de bursary fund om mensen die het niet kunnen betalen toch de gelegenheid te geven aan euroT_EX deel te nemen.
- **Administratie** Kosten die gemaakt worden door de penningmeester en secretaris voor het versturen van allerlei verenigingspost en verzoeken om inlichting van potentiële nieuwe leden. Dit bedrag valt wel lager uit dan begroot maar komt voornamelijk omdat veel post in combinatie met MAPSen worden verstuurd en dan de gehele rekening wel ten kosten van de MAPS gerekend wordt.

- **KvK en notaris** Jaarlijkse kosten en de kosten van een extra uittreksel als gevolg van de bestuurswisselingen.
- **Bestuurskosten** Zo af en toe vergadert het NTG bestuur om een aantal zaken die niet via e-mail afgehandeld kunnen worden te bespreken (dit vergaderen is soms telefonisch maar soms ook op locatie). Ook staan hier kosten die het bestuur maakt i.v.m. contacten met andere usergroups.
- **Mededelingen aan de leden** Hieronder valt de MAPS maar ook andere post. Uit de begroting blijkt dat we meer hebben uitgegeven dan begroot. Dit is in eerste instantie het directe gevolg van meer leden dan was begroot. Als we weten dat er f1.200,00 gulden aan reclame inkomsten zijn, f357,50 aan verkoop van oude MAPSen en een 4all \TeX /euro \TeX sponsoring van f2.815,68 voor de dure proceedings, dan moeten we concluderen dat er wel degelijk veel meer is uitgegeven aan deze post dan is begroot en geeft dit de verklaring voor een negatief saldo.
- **Reisbijdragen** Dit jaar zijn alle reisbijdragen (Polen, Amerika) betaald door 4all \TeX .
- **Verkoop** In 1995 is een aparte giro rekening geopend voor de CD-rom en zijn de meeste transacties via deze rekening verlopen. Het saldo van de verdiensten van 1994 (f14.187,88) zijn op deze giro rekening overgemaakt. De creditcard aanvragen worden nog via de gewone NTG rekening afgehandeld (goed voor een f7.164,92 wat te vinden is op de balans als post crediteuren).
- **Afschrijvingen** De NTG heeft een PostScript printer welke in 3 jaar wordt afgeschreven.
- **Saldo** Het lijkt een negatief saldo maar als we de CD-rom inkomsten (die geen eigendom van de NTG zijn) er van aftrekken dan hebben we dit jaar een **positief** saldo van f2.521,70. Dit lijkt een prima resultaat, maar dan vergeet men dat eigenlijk de CD-rom en de MAPS sponsoring voor dit positief saldo zorgen (de reisbijdragen van f2.000,- zijn voor rekening van de CD-rom, een sponsoring van f2.816,68 voor de euro \TeX maps en een f1.200,- aan MAPS sponsoring zouden anders geleid hebben tot een verlies van f3.478,30).

3 De balans

In tabel 3 is de balans per 31 december 1995 weergegeven en in tabel 4 is de balans per 1 februari 1995 weergegeven.

4 Conclusie

De 4all \TeX CD-rom zorgt er voor dat dit jaar niet rampzalig is verlopen. Er zijn dan ook twee mogelijkheden voor de toekomst. Of we verhogen de contributie (om 1995 sluitend te krijgen had dat ongeveer f20,00 per lid moeten zijn) of we stellen strikte limieten aan de MAPS (uitgaven van maximaal 175 pagina's en geen MAPS specials). Verder dient de NTG er over na te denken wat men met 4all \TeX wenst te doen. Het produceren van een CD-rom en alle bijkomende zaken (verkoop, onderhoud, support etc.) vergt dusdanig veel energie dat het niet aan een enkeling kan worden overgelaten.

<i>Bedragen in guldens</i>	Aktiva	Passiva
Giro	17.295,38	
Kas	100,00	
Contributies	1.510,00	714,96
Debiteuren	79,74	
Crediteuren		7.407,42
Kapitaal		12.044,79
Apparatuur	3.546,15	
Afschrijving		2.364,10
	22.531,28	22.531,27

Tabel 3: De balans per 31 december 1995

<i>Bedragen in guldens</i>	Aktiva	Passiva
Giro	22.095,54	
Kas	532,70	
Contributies	671,00	
Debiteuren	450,94	
Crediteuren		2.403,31
Kapitaal		22.528,93
Apparatuur	2.364,10	
Afschrijving		1.182,05
	26.114,29	26.114,29

Tabel 4: De balans per 1 februari 1995

Jaarverslag NTG

jan–dec 1995

Gerard van Nes

1 Algemeen

Het jaar 1995 was voor de NTG een zeer druk en wederom succesvol jaar. Toppers waren natuurlijk EuroTeX'95 en de uitgave van de derde editie van de 4allTeX CD-rom. Twee NTG-bijeenkomsten (waarvan de eerste goed bezocht) vonden plaats met vele goede lezingen van sprekers van zowel binnen als buiten de NTG. Er verscheen een tweetal uitgebreide MAPS uitgaven (Minutes & Appendixes). Daarnaast werd begin 1995, de MAPS'94 Special Edition aan alle NTG-leden beschikbaar gesteld.

En voor de rest is er het diverse door de leden gedaan waaronder onder meer het beheer van het Bulletin Board FGBBS, het beheer van de TEX-NL en de uitgebreide hulpverlening bij vragen op de nationale TEX-NL en 4TeX discussielijst. En niet te vergeten de medewerking aan het L^AT_EX₂ ϵ project.

Helaas wordt het vele NTG-werk door slechts een *te beperkte groep* actievelingen uitgevoerd. Er blijkt duidelijk behoefte te zijn aan uitbreiding...

2 Het NTG bestuur

Het NTG bestuur bestond in 1995 uit de volgende personen:

- J.L. Braams, voorzitter (tot juni 1995),
- J.L. Braams, bestuurslid (vanaf juni 1995),
- E.H.M. Frambach, voorzitter (vanaf juni 1995),
- E.H.M. Frambach, bestuurslid (tot juni 1995),
- G.J.H. van Nes, secretaris,
- W. Dol, penningmeester,
- F. Goddijn, bestuurslid,

Op de NTG-bijeenkomst in juni vonden de bestuursverkiezingen plaats. Erik Frambach en Gerard van Nes traden reglementair af. Beiden stelden zich echter herkiesbaar en werden vervolgens bij acclamatie herkozen. Johannes Braams droeg de voorzittershamer over aan Erik Frambach.

Philippe Vanoverbeke continueerde ook in 1995 zijn functie van NTG België-commissaris en verstevigde daarmee de contacten met onze Vlaamse leden.

3 Het NTG ledenbestand

Het ledenaantal bleef ook weer in 1995 stijgen, echter wel duidelijk minder dan in de jaren daarvoor. Het aantal instituutleden nam daarentegen af. Eind 1995 telde de NTG 286 leden waarvan 33 instituutleden. Eind 1990/1991/1992/1993/1994 bedroeg het aantal leden res-

pectievelijk 117/159/193/232/265 en het aantal instituutleden 21/28/30/33/36. Mede dankzij MAPS en 4allTeX CD-rom activiteiten lijkt het ledenaantal in 1996 de 300-grens te gaan passeren.

De contributie werd in 1995 voor het eerst — sinds de oprichting van de vereniging in 1988 — met *f* 15,- per individueel lidmaatschap verhoogd hetgeen mede werd veroorzaakt door de toegenomen MAPS- en algemene portokosten.

De samenwerking van NTG met TUG — waarbij het lidmaatschap gecombineerd kon worden overgemaakt — bleek te resulteren in 85 NTG/TUG leden eind 1995 (83 eind 1994). Vanwege de verlate toezending van de het TUG-tijdschrift 'TUGboat' werden enkele gecombineerde lidmaatschappen omgezet in NTG-lidmaatschappen.

Vanwege de omvangrijke MAPS (en EuroTeX'95 proceedings) uitgaven werd in 1995 geen volledige NTG-ledenlijst gepubliceerd (dit zal begin 1996 plaatsvinden).

Opvallend was eind 1995 het aantal NTG-leden (orde grootte 20) die, ondanks diverse aansporingen, nog niet aan de contributieverplichtingen over het lopende jaar hadden voldaan.

4 De NTG bijeenkomsten

Er zijn in 1995 ook weer twee NTG-bijeenkomsten georganiseerd welke gekenmerkt werden door een levendige discussie en een hoge mate van informatie-uitwisseling.

1. Op 24 mei 1995 bij de Universiteit Twente. Aanwezig waren 56 leden.
2. Op 5 september 1995 tijdens de EuroTeX'95 bijeenkomst op Papendal te Arnhem. Aanwezig waren 33 leden.

Op de eerstgenoemde bijeenkomst met als thema: 'Multimedia en TeX'. vonden de volgende lezingen plaats:

- *De NTG op het World Wide Web* door Henk de Haan;
- *L^AT_EX₂HTML ervaringen* door Arno Kemperman (Katholieke Universiteit Nijmegen);
- *Xhdvi ervaringen* door Peter Gragert (Universiteit Twente);
- *Adobe Acrobat 2.0; 'beyond the bounds of paper'* door Wiegert Tierie (Adobe Systems);
- *Het produceren van intelligente teksten met TeX* door Hans Hagen (Pragma).

Deelnemers aan de najaars NTG-bijeenkomst konden op dezelfde dag vrijblijvend de lezingen (met thema 'Graphics

and packages' en 'Electronic documents') van EuroTeX'95 volgen.

Van bovengenoemde twee vergaderingen (en lezingen) verschenen verslagen met bijlagen in MAPS 95.1 en MAPS 95.2. Een deel van de lezingen werd gepresenteerd door sprekers die door de NTG speciaal voor deze gelegenheden waren uitgenodigd.

En natuurlijk kwam ook weer het nodige over 4TeX (en 4allTeX) weer ter sprake met als vaste sprekers Wietse Dol en Erik Frambach.

De leestafels met TeX-achtige boeken, vorige uitgaven van de MAPS, diverse LUG- en andere tijdschriften, brochures en diverse aan TeX verwante documenten, trok zoals altijd de nodige belangstelling. Bij zowel de voorjaars- als de naajaarsbijeenkomst konden de aanwezigen Addison-Wesley (IA)TeX/PostScript boeken met reductie aanschaffen.

5 EuroTeX'95

De Europese TeX bijeenkomst EuroTeX vond in 1995 voor het eerst in Nederland plaats op het sport- en congrescentrum Papendal te Arnhem. Gedurende een week vonden ruim 30 lezingen en een vijftal tutorials plaats. Het aantal deelnemers lag tegen de honderd. Zeer opvallend was echter de sterke absentie van de NTG leden. Slechts een 15-tal (inclusief bestuursleden) waren aanwezig ondanks een eerdere peiling tijdens een NTG-bijeenkomst welke een schatting gaf van ongeveer 75 NTG-deelnemers.

Uniek voor deze bijeenkomst was de 'Oost-Europa TeX-bus' die 50 Oost- en Mid-Europa TeX-ers van en naar Warsau bracht. De organisatie was in handen van Kees van der Laan met hulp van de betrokken LUGs: GUST (Julita Bolland, Tomek Przechlewski), CyrTUG (Irina Makhovaja) en CS-TuG (Petr Sojka). Een fraai staaltje van samenwerking. Door de (gesponsorde) bus waren de congreskosten voor deze Oost- en Mid-Europa TeX-ers laag.

De social event bestond uit een boottocht op de Rijn inclusief diner, gearrangeerd door Theo Jurriens. Het nationale Park 'De Hoge Veluwe' gaf m.n. de Russen de gelegenheid te (leren) fietsen op onze witte fietsen, naast het genieten van een van onze mooiste kunstschaten in een uniek park. Informeel werd op vrijdag een dagtripje georganiseerd naar Amsterdam door Kees van der Laan, die tevens voorging op een wandeling door de stad met als eerste stop de winkel van Computer Collectief. De bloemenmarkt was in trek. De komende BachoTeX zal hierdoor opgefleurd worden met Hollandse tulpen.

De coördinatie van EuroTeX'95 was in handen van Erik Frambach. Hij werd daarbij ter zijde gestaan, naast de NTG-bestuursleden, door onder meer Theo Jurriens, Ruud Koning, Kees van der Laan, Piet van Oostrum Simon Pepping en Piet Tutelaers.

Vele sponsors maakten EuroTeX'95 mogelijk. De *hoofdsponsor* van de conferentie was 4allTeX. Daarnaast werd sponsoring geleverd door Elsevier, Wolters Kluwer, Adobe, Computer Collectief, Scientific Word, Sun, Y & Y, Addison-Wesley, TUG'95 (de boekenveiling), en ver-

schillende TeX Users Groups. Dankzij deze sponsoring kon EuroTeX'95 zonder verlies (maar wel met een mentale winst) afgesloten worden.

Vóór de aanvang van de conferentie was een 440 pagina's dikke proceedings beschikbaar. Wietse Dol wist in zeer korte tijd dit unieke boekwerk (welke als bijlage bij MAPS #15 aan de NTG-leden werd verstuurd) te realiseren. Algemene positieve waardering was er over de vormgeving van dit unieke boekwerk.

6 De NTG MAPS

Ook in 1995 verschenen er twee MAPS uitgaven met ruim 235 goed gevulde bladzijden *plus* de EuroTeX'95 proceedings met ruim 440 (!!) bladzijden. Daarbij onder meer als inhoud: verslagen van nationale en internationale bijeenkomsten, artikelen m.b.t. de NTG lezingen, zeer veel tutorial- en ander educatiemateriaal, TeX/LaTeX & PostScript bijdragen, font-, TeX- en PC-, hypertext-artikelen, bijdragen m.b.t. TeX en LaTeX macro's, de laatste informatie over LaTeX3, SGML en algemene plus technische TeX en LaTeX bijdragen.

De MAPS-redactie bestond in 1995 uit Wietse Dol en Gerard van Nes, met redactionele medewerking van Frans Goddijn, Philippe Vanoverbeke en Jos Winnink. MAPS 95.1 werd ook *zowel in PostScript als PDF* beschikbaar gesteld op de derde editie van de 4allTeX CD-rom.

Begin 1995 werd aan alle NTG-leden, de MAPS'94 Special Edition toegestuurd. Basis voor deze MAPS-special waren de '100 Frequently Asked Questions' uit het tijdschrift van de UK TeX Gebruikersgroep 'Baskerville'.

In juni 1995 is, in samenwerking met Christina Thiele, de (besloten) internationale TeX-eds discussielijst opgericht t.b.v. de editors van de locale gebruikersgroep tijdschriften.

Eind 1995 is de *eindredactie* van de MAPS overgegaan van Gerard van Nes naar Wietse Dol. Eerstgenoemde blijft wel actief binnen de redactie.

7 De NTG werkgroepen

Ook gedurende 1995 blijken de NTG-werkgroepen te zijn ingesluiemd. Activiteiten komen hoofdzakelijk van individuele leden. De veel gebruikte TEX-NL discussielijst zorgt reeds dikwijls voor een vorm van 'overleg'.

8 De NTG CD-ROM

De belangstelling voor de 4allTeX CD-rom bleef ook in 1995 toenemen. Op PC-gebied is de CD-rom inmiddels internationaal een standaard. In juli kwam de derde editie van CD-rom beschikbaar. Een *dubbel*-CD-rom. Net op tijd om de eerste exemplaren ervan tijdens de TUG'95 bijeenkomst in Florida uit te reiken. Donald Knuth ontving het eerste exemplaar. De gold disc van de derde editie werd overhandigd aan TUG.

Veel werk werd in het gehele CD-rom ontwikkel- en productieproces verzet door het koppel Wietse Dol & Erik

Frambach, doch ook diverse mensen er omheen hebben het nodige bijgedragen aan het succes.

9 NTG op World Wide Web

NTG was ook in 1995 op het WWW beschikbaar. Acties zijn ondernomen om de WWW-server een definitieve plaats te geven. Diverse pagina's zijn echter sterk achterhaald. Een grondige update wordt voor 1996 gepland.

Het adres is: <http://www.ntg.nl/>.

Het nodige t.b.v. de NTG WWW-server is gedaan door onder meer Piet van Oostrum en Jules van Weerden. Eerstgenoemde beheert thans de NTG WWW-site en is bezig een volledige CTAN-mirror op te zetten op nieuwe hardware die de NTG uit de opbrengst van de 4all \TeX CD-rom heeft bekostigd.

10 Bestuursactiviteiten

De NTG wisselde informatie uit met de LUGs: Exemplaren van de MAPS werden verstuurd naar de meeste zusterorganisaties (inclusief TUG). Van de Duitse en Franse gebruikersgroep ontving de NTG de periodieken (zie ook de leestafel tijdens NTG-bijeenkomsten!).

Namens de NTG werden in 1995 de volgende bijeenkomsten bezocht:

- **TUG'95 te St. Petersburg, Florida.**

Van 24 – 28 juli werd door de secretaris en de penningmeester deelgenomen aan deze jaarlijkse TUG meeting. Een 4 \TeX -lezing werd aldaar gegeven en de TeX-editors bijeenkomsten werden bijgewoond. Een uitgebreid verslag van deze bijeenkomst is gepubliceerd in MAPS #15 (blz 8 –13).

- **EuroTeX'95 te Arnhem.**

Natuurlijk was het gehele bestuur (op persoonlijke titel) bij deze belangrijke conferentie van 4 – 8 september 1995 aanwezig. Daarnaast waren gedurende deze week een handvol andere NTG-leden aanwezig. Op de tweede dag van deze bijeenkomst werd tijd vrijgemaakt voor de 16^e NTG-bijeenkomst.

- **BachoTeX'95 te Polen.**

Van 29 april tot 3 mei werd eveneens door Erik Frambach en Kees van der Laan deze Oost-Europese GUST'95 bijeenkomst bijgewoond. Lezingen m.b.t. 4all \TeX en BLUe's zaken werden gegeven.

Een verslag van deze bijeenkomst is in MAPS 95.1 (blz. 212–213) opgenomen.

Op 8 maart en 26 augustus vonden *bestuursvergaderingen* plaats waarin naast de algemene gang van zaken binnen de NTG, onder meer de organisatie van de NTG bijeenkomsten, Public Relation activiteiten, 4all \TeX (distributiemogelijkheden), financiële zaken, royementen, Euro \TeX , taakverdeling binnen het bestuur en de gebeurtenissen binnen TUG en andere zusterorganisaties ter sprake kwamen.

11 Diversen

- Van de TEX-NL discussielijst werd in 1995 wederom veel gebruik gemaakt. Zowel het aantal abonnees van deze listserver als ook het e-mail verkeer bleef tamelijk stabiel. Aantal abonnees eind 1992/1993/1994/1995: 162, 180, 175 en 181. Aantal e-mails gedurende 1992/1993/1994/1995: 1097/1047/1242/1216.

Boven aan de lijst van TEX-NL actievelingen (op een hoge eenzame plaats) stond Piet van Oostrum die met 140 verstuurd e-mails (nagenoeg 100% hiervan waren antwoorden op gestelde vragen), veel \TeX gebruikers uitstekend wist te ondersteunen.

De activiteiten van de TEX-NL fileservers waren daarentegen minimaal, vooral vanwege aanwezige alternatieven.

- De in april 1994 gestarte 4TEX discussielijst kende eind 1995 232 abonnees wereldwijd (eind 1994: 170). Deze lijst bleek ook duidelijk in een behoefte te voorzien.

Er werden in 1995 in het totaal 529 e-mails verstuurd (in 1994: 572) met als top-3 lijst van verstuurders wederom: Wietse Dol, Erik Frambach en Frans Goddijn. Ze namen samen ongeveer 59% (!) van de e-mails voor hun rekening, in nagenoeg alle gevallen voor het beantwoorden van allerlei soorten 4 \TeX en 4all \TeX achtige vragen.

- Het NTG-FGBBS *Bulletin Board* van Frans Goddijn werd door zowel NTG-ers als niet NTG-ers nog steeds dagelijks bezocht. Het FGBBS bevat naast de 4all \TeX CD-ROM, een indrukwekkende hoeveelheid aan (L^A) \TeX materiaal.
- Vele leden maakten via e-mail dan wel via ftp dankbaar gebruik van de *RUU fileserver* van Piet van Oostrum voor het verkrijgen van diverse soorten \TeX materiaal. Daarnaast werd ook gebruik gemaakt van enkele andere ftp-server systemen in Nederland.
- Actieve medewerking werd verleend aan het L^A \TeX 2 ϵ project door het L^A \TeX 3 teamlid Johannes Braams.
- Wietse Dol toonde tijdens TUG'95 conferentie in St. Petersburg de mogelijkheden van 4all \TeX zowel in de vorm van een lezing als in de vorm van een cursus. Op de Euro \TeX '95 conferentie in Arnhem werd door Johannes Braams de status van babel uiteengezet. Kees van der Laan gaf presentaties over het indexen in onepass en het gebruik van de database functionaliteit binnen BLUe's format. Hans Hagen vertelde over de markup van chemische formules. Erik-Jan Vens leverde een bijdrage over conversie van Euler METAFONT naar PostScript Type1. Andries Lenstra verhaalde over zijn $\delta\alpha\TeX$.

Tijdens deze conferentie gaf Piet van Oostrum een tutorial over de nieuwe FANCYHEADINGS mogelijkheden.

Van de TuG president

Michel Goossens

CERN, Geneva, Switzerland
goossens@cern.ch

De winter ligt eindelijk achter ons, het bleef lang koud, maar de lentezon is er weer en zendt ons haar gulle warme stralen. Dra is 't weerom zomer en in ben ervan overtuigd dat velen al sinds enige tijd plannen smeden voor de vakantie, alleen, met vrienden, of met de familie.

TUG heeft ook een “vries”-periode doorgemaakt, met een monotoon aftakelen van het ledental, en de daarmee gepaard gaande financiële moeilijkheden. Om te trachten de kop boven water te houden (letterlijk !) beloofde ik begin augustus in mijn toespraak in Florida op TUG96 dat mijn eerste prioriteit als inkomend voorzitter het publiceren van alle achterstallige nummers van TUGboat zou zijn, gezien het ontvangen van *het* referentietijdschrift van de \TeX -wereld het enige is wat de meeste TUG leden daadwerkelijk ontvangen voor hun bijdrage. Dit betekende dat we in vijf maanden, van september 1995 tot januari 1996, evenveel nummers van TUG's in principe trimestrieel blad moesten samenstellen, drukken, en versturen aan alle leden. Dat kwam overeen met wel over de 500 bladzijden. Om financiële redenen hebben we eind vorig jaar ook het uitgeven van TTN (*TeX and TUG News*) stopgezet. Ik denk dat deze beslissing niettemin een logische stap is in het concentreren van onze inspanningen om van TUGboat *het* internationale referentieblad van de \TeX gemeenschap te maken. Maar over de herstructurering van TUG meer later.

Zoals gezegd zijn alle achterstallige nummers van 1994 and 1995 sinds eind januari of begin februari van dit jaar bij alle TUG leden toegekomen—ik weet dat er enkele problemen waren bij laat-vernieuwers, die tot in het najaar, toen ze eindelijk TUGboat 15#4, het laatste nummer van 1994, kregen, wachtten om hun 1995 bijdrage te storten. Dit bracht met zich mee dat hun naam niet op de eerste ledenlijst van 1995 voorkwam—we hebben dit probleem nu weer met de laat hernieuwende 1996 leden!—zodat ze TUGboat 16#1 niet onmiddellijk ontvingen. Ik hoop dat deze problemen intussen zijn opgelost, maar, gezien we in het TUG kantoor nog slechts over één persoon beschikken, die zich met alles moet bezighouden, kan het wel eventjes duren voor er kan gereageerd worden op elektronische of gewone mail. Ik vind het natuurlijk spijtig dat we in zo'n ietwat akelige positie zijn terecht gekomen, maar we kunnen nu eenmaal de klok niet terugdraaien, en ik wil bijgevolg pleiten voor wat begrip in de huidige toestand. Ik hoop op de positieve en actieve medewerking van alle MAPS lezers te kunnen rekenen om TUG er zo vlug mogelijk weer bovenop te helpen.

Ondertussen zijn we er dank zij het harde werk van het *TUGboat Production Team* in geslaagd, voor het eerst sinds 1992, TUGboat weer op tijd te laten verschijnen, d.w.z. het maartnummer is in maart naar de drukker gegaan. Ik verwacht dat de huidige TUG leden dit eerste nummer van 1996 (17#1) nog voor eind april in hun bus krijgen, en dat dit feit vele MAP lezers (en NTG leden, dus geestdriftige \TeX fans) ertoe zal aanzetten hun bijdrage voor 1996 naar NTG te zenden. Laat me deze gelegenheid nemen om het NTG bestuur te danken voor hun jarenlange steun aan TUG, in het bijzonder voor het gemeenschappelijk lidmaatschap NTG-TUG, waar de leden van onze beide organisaties een maximaal voordeel uit halen.

Het past tevens hier mijn medewerkers van het bovengenoemde *TUG Production Team* eventjes in de lentebloementjes te zetten. Het Team bestaat uit Barbara Beeton van de AMS, de verantwoordelijke uitgever van TUGboat, Mimi Burbank van Florida State University, de productie-leider, Robin Fairbairns van Cambridge en UKTUG, Sebastian Rahtz, van Elsevier Science in Oxford, TUG en UKTUG, Christina Thiele, de vorige presidente van TUG, en ikzelf. Voor de twee laatste nummers konden we op de medewerking rekenen van Wietse Dol, die ons enkele artikels van Euro \TeX '95 doorspeelde, die door de respectievelijke auteurs, indien nodig, ietwat werden geactualiseerd, en Malcolm Clark, die als eindredacteur optrad voor TUGboat 16#2, het speciale nummer over elektronische documenten.

Waar staat TUG nu begin 1996 ? Het spijt me te moeten melden dat de toekomst er niet al te rooskleurig uitziet. Eind maart hadden iets meer dan twaalfhonderd leden hun lidmaatschap hernieuwd. Dat is een derde minder dan eind 1995. Er bestaat nog goede hoop dat we het ledenaantal van vorig jaar zouden kunnen benaderen nu TUGboat weer regelmatig verschijnt, maar om TUG min of meer rond te laten komen in zijn huidige vorm hebben we wel meer dan tweeduizend leden nodig, en dat zit er waarschijnlijk niet meer in. Gezien TUG over wat financiële reserves beschikte konden we vorig jaar nog eventjes verder, maar voor het einde van dit jaar zullen we moeten overgaan op een structuur zonder permanent personeel of bureel. Dit betekent dat TUG zich zal concentreren op het uitgeven van TUGboat, een publicatie die we meer dan ooit tevoren als *het* focuspunt can alle \TeX activiteiten wereldwijd zouden willen doen uitgroeien. Alle \TeX gebruikersgroepen zullen in TUGboat met hun aankondigingen terechtkun-

nen en we hopen om, via een versterkte samenwerking tussen de eindredacties van de verschillende \TeX bladen, alle \TeX gebruikers over de hele wereld te laten profiteren door het herpubliceren en, indien nodig, in het Engels vertalen, van de beste inleidingen, tutorials, en referentieartikels over \TeX , MetaFont, en hun vriendjes.

Sinds begin april heeft TUG aan de universiteit van Massachusetts (umb) zijn eigen plaats op de Internet (een PC die Linux draait), en we hebben er een web server (<http://www.tug.org/>) en een ftp server (<ftp.tug.org>). Het is Karl Berry (de auteur van, o.a., de beroemde en veelgebruikte web2c \TeX -versie) die TUG de PC ter beschikking stelde en ons in de mogelijkheid stelde hem kosteloos via de universiteit waar hij werkzaam is op de Internet aan te sluiten. Karl en het TUGboat Production Team hopen die `www` en `ftp` servers tot een zo compleet mogelijke bron van informatie over \TeX uit te bouwen, met ondermeer veelvuldige wijzigingen naar andere informatiebronnen en wat andere \TeX gebruikersgroepen aan te bieden hebben. En, om nogmaals aan te tonen dat TUG de elektronische toer opgaat, gaan we begin juni, samen met GUTenberg en UKTUG,

een “plug en play” CDROM voor Unix machines op de markt brengen. De structuur is gebaseerd op de TDS (\TeX Directory Structure) en op de `tetex` distributie van Thomas Esser (een quasi-automatische en geoptimaliseerde installatie procedure voor Karl Berry’s `web2c`). Er zullen uitvoerbare modules voor een twintigtal verschillende hardwareconfiguraties opstaan¹. Te koop in alle goede \TeX -winkels (ook bij NTG !) vanaf juni.

Ik hoop dat ik iedereen ervan overtuigd heb dat TUG, na een tamelijk lange verkoudheid, aan de beterhand is. Nochtans moet iedereen, d.w.z. alle \TeX -gebruikers ter wereld, i.h.b. de NTG leden die dit lezen, ons helpen in deze belangrijke consolidatieperiode. TUG (zoals NTG, GUTenberg, UKTUG, enz.) zal zich moeten aanpassen aan de nieuwe informaticaomgeving, waar het niet langer papier, magneetbanden, mainframes en diskettes zijn die de hoofdrol spelen, maar elektronische documenten in `pdf`, `PostScript`, `LATEX` of `SGML` formaat, en waar de Internet en CDROMs immer belangrijker worden als informatieverbreidingsmedia. Ik hoop vanharte op jullie medewerking te kunnen rekenen.

¹Begin maart produceerde ik samen met Sebastian Rahtz bij Elsevier Science in Oxford twee “gouden” CD’s. Om de bestanden grondig te testen heb ik alles voor negen verschillende architecturen op CERN op afs gezet. Tevens voer ik de tekst van dit artikel in op mijn PC die Linux draait. Ik gebruik mijn meesterCD om \TeX te draaien zonder een enkel `LATEX` bestand (behalve mijn tekst, natuurlijk) op mijn harde schijf te hebben. En dank zij het stuurbestand dat de \TeX boomstructuur beschrijft, gaat het vlugger van de CD dan van de harde schijf op een Sun of HP.

Bericht van FGBBS

Frans Goddijn

1 Heeft FGBBS nog toekomst?

Deze vraag stelde ik me het afgelopen jaar regelmatig. Van twee kanten zag ik reden tot twijfel...

Ten eerste heeft mijn cosytop Henk de Haan, sinds hij zijn studie heeft afgerond met een voortreffelijk (natuurlijk in L^AT_EX vormgegeven) proefschrift, minder tijd voor FGBBS. Henk heeft een drukke baan in de ontwikkeling van *smart cards* en bovendien wordt er in zijn huidige werkomgeving niets met T_EX gedaan. De tijd dat hij, bij vlagen dag en nacht, bezig was om nieuwe files voor FGBBS op te halen, of nieuwe *features* te bedenken voor de bellers van het systeem, is dus voorbij. Dit betekent dat FGBBS vrijwel verstoken blijft van de nieuwste files. De regelmatige verschijning van de 4T_EX CD compenseert dit laatste enigszins: deze is op FGBBS geplaatst en bovendien kan iedereen de (dubbel-)CD voor een prikje in huis halen. Wel is het zo dat de technische basis wat is versmald: zo duurde het vrij lang eer ik tegen het einde van het jaar door had dat een aantal bellers FGBBS niet meer binnen kon komen, en vervolgens had ik er veel moeite mee om uit te zoeken hoe dit modemprobleem kon worden verholpen. Overigens is het nog steeds zo dat de kern-software van FGBBS nog steeds zo draait als Henk de Haan het ooit heeft geïnstalleerd, inclusief de link naar het Internet. Hoe dan ook heb ik, en de bellers met mij, veel aan Henk te danken.

Internet brengt me op de tweede en even voorname reden om aan de bestaansredenen voor FGBBS te twijfelen: nu iedereen die dat per se wil, en die er ook geld voor (over) heeft, het Internet kan bereiken, is het minder noodzakelijk om alternatieven te bieden. Wie een Internetaccount heeft, kan ook alle T_EX-berichten lezen, kan de nieuwste T_EX-files opzoeken en kan de homepage van Donald Knuth bezoeken. Waarom zou er dan nog een BBS moeten zijn die nog eens datzelfde biedt?

Ik wierp deze vraag op bij een aantal modemvrienden en zij antwoordden als volgt.

Gerhard Hoogterp (gerhard@frappe.iaf.nl) vertelde dat het een discussie is "die ik voor mijn (bedrijfs/professionele) BBS (053-4324909) regelmatig moet voeren. Persoonlijk denk ik dat BBS-en als zodanig blijven bestaan. Punt is natuurlijk hoe je een BBS definieert. Op dit moment is het zo dat op Internet bijna alles open ligt. Iedereen kan overal bij en de hoeveelheid data is overweldigend. Helaas betekent dat ook dat het vinden van data niet echt eenvoudig is. Echt slimme zoekengines zijn er nog amper. Om to-the-point informatie over een bepaald on-

derwerp te krijgen, moet er een index gemaakt worden. Het maken van professionele indexes kost geld. Het kan niet zo zijn dat het overgrote deel van Internet gratis blijft zoals dat nu min of meer het geval is. Op een gegeven moment zal dus voor het gebruik van bepaalde zoeksystemen betaald moeten gaan worden. Betaling is gekoppeld aan autorisatie. Heb je eenmaal een groep mensen met een soortgelijke interesse bij elkaar, dan is een mailinglijst of een setje lokale newsgroups ook snel gerealiseerd. Heb je eenmaal berichten, informatie (en er zal vast ook wel een FTP server met gespecialiseerde files beschikbaar komen) dan heb je toch iets wat verdacht veel op een BBS begint te lijken."

"Als je nu al een (gespecialiseerd) BBS hebt," zo redeneert Hoogterp, "dan kun je dus net zo goed eens gaan kijken of je dat BBS niet met Internet kunt intergreren. BBS-en met algemene files zullen verdwijnen. Wie kan er concurreren met <ftp.funet.fi> etc. Maar een BBS als het mijne (SLO-Lijn) met specifieke informatie voor het nederlands onderwijs kan volgens mij nog prima een tijdje meedraaien in een hybride vorm. Eerst bereikbaar via telnet/vmodem zodat gebruikers tegen lokaal tarief, via hun internet verbinding *kunnen* inloggen, stap twee om bepaalde gedeeltes van het BBS via b.v. CGI modules en het WWW ook bereikbaar te maken. Hierbij gaat het dan overigens wel over een gratis service. Wat mij betreft zie ik de toekomst van BBS'en als: Het BBS is dood, lang leve het BBS!"

Frans van Geene (Frans_van.Geene@gemini.iaf.nl) ziet nog wel degelijk een toekomst voor het BBS: "Als je je kunt richten op bepaalde doelgroepen, en als de hele hausse van het nieuwe van Internet eraf is, keren de mensen vanzelf wel weer terug naar het vertrouwde BBS-op-de-hoek. Wanneer je een BBS kan zien als intermediair voor de communicatie (E-Mail/News/Netmail/Echomail) dan blijft het voor zeer veel mensen een aantrekkelijk alternatief. Er zijn nog genoeg bellers van BBS-en waarvoor Internet 'te hoog gegrepen' is, of waarvoor de investering in andere hard- en software en een abonnement te duur is. Bij mij is de grootste loop er ook uit, maar is het aantal vaste bellers en lezers/schrijvers van berichten toegenomen. Liever een kleine 'vaste' kern dan 100 tijdelijke bellers die alleen maar op zoek zijn naar *plaatjes*..."

Gregor Tee (Gregor@idn.nl) echter verwacht dat de prijzen snel zullen kelderden, ook als we straks Internet via de kabel krijgen. "Ik ga m'n BBS in ieder geval binnenkort helemaal opdoeken, misschien dat er een klein ftp-siteje over blijft, maar dat moet ik nog bekijken." (bbs/fax 050-3133623)

Jan Zegeren (Jan@zsoft.iaf.nl) vermoedt dat de BBS-en "op dit moment nog bestaan omdat de drempel en de kosten voor een Internetverbinding veel hoger zijn dan het lokaal even inbellen om een bestandje te downloaden of een berichtje te lezen. Daarnaast is de huidige softwaregeneratie gewoonweg nog te ingewikkeld. Het moge onder-tussen duidelijk zijn dat e-mail veel sneller en eenvoudiger werkt dan het reguliere fax- en briefverkeer. Het is niet voor niets dat steeds meer bedrijven zich laten aansluiten op Internet. Ook zie je duidelijk dat alle BBS-en met hun filenetten alle files naar elkaar overpompen (wellicht om de PTT te sponsoren). De mailareas van een fido-BBS als zodanig zijn voor een beginnende modemgebruiker makkelijker toegankelijk dan een newsgroup, waar je alweer een account voor moet hebben."

"Het grootste voordeel", aldus Zegeren, "van de mailareas op een BBS is natuurlijk dat het op een relatief eenvoudige manier offline thuis te beantwoorden is (is goedkoper, dus erg belangrijk in ons eigen Nederland). Nu klopt het natuurlijk als je zegt dat je dat met UUCP ook kunt, maar de meest onvriendelijke en ontoegankelijke software die op mijn systeem staat, is toch wel de UUCP software. Op het moment dat Internet betaalbaarder is en je meer zaken op een eenvoudige manier offline kunt gaan regelen, en dit geheel verpakt in een vriendelijk, goed ogend stukje software, dan is het einde van de vele BBS-en echt in zicht."

Geert Bosch (geert@sun3.iaf.nl) tenslotte schreef:

Toch heb ik in het verleden een *hele hoop* software bij jou opgehaald om alle TeX-spullen aan het werk te krijgen. Als je dan met veel zoeken op het internet weer het een en ander gevonden had, miste je wel weer het ander. Ook zaken zoals de Nederlandse stijlen en de nieuwste Babel-versies waren wel erg handig. En toen Latex2e kwam, wist ik ook wel waar ik alle nieuwe spul kon vinden!

Nog hartelijk bedankt voor het mogen gebruiken van je BBS!

Groeten,
Geert

PS We gebruiken LaTeX hier nog dagelijks voor onze afstudeerscripties, op een tien jaar oude Sun 3/60 en grote 20"monitor ;-)

Ik heb besloten om FGBBS (026-3217041) toch nog maar een poosje online te houden. BBS-en vormen een goedkoop alternatief en je kunt er op lokaal of nationaal niveau nog veel mee betekenen, vooral door specialisatie. Informatie op mijn Internet homepage kan mensen naar FGBBS leiden en andersom.

Wel vind ik in de Internetontwikkelingen aanleiding om dit jaar niet te vragen om een NTG-subsidie. Als er geld over is, kan dit volgens mij beter aan een NTG-site op Internet worden besteed. Wel overweeg ik om voor het FGBBS van de bellers een donateursbijdrage te vragen, waar ik dan meer bevoegdheden (voor files en email) tegenover kan stellen.

Tussen 09-13-94 en 04/14/96 (579 dagen) is er 1778 keer iemand op bezoek geweest en er zijn 637 bellers met naam en wachtwoord ingeschreven. Er komen dus gemiddels zo'n drie bezoekers per dag langs. Heel soms laat er iemand een berichtje achter, zoals eergisteren:

```
Date : Fri Apr 12, 22:17
From : Bernard Van Gastel
To   : Frans Goddijn
Subj : bbs
```

Hoi Frans

Mooi BBS is dit, alleen ik weet niet wat TEX is? Kun je het me even E-MAILEN via INTERNET?

Bernard!
^^^^^^^

E-MAIL : GASTEL@WORLDACCESS.NL

P.S. Bedankt!

Natuurlijk heb ik Bernard via Internet informatie toegestuurd over TeX, over de NTG, over Donald Knuth en over de komende NTG-dag. Een mooi voorbeeld van de rol die een BBS en Internet beide kunnen spelen, is 't niet?

A plug-and-play teTeX CD-ROM

Michel Goossens and Sebastian Rahtz

GUTenberg and UKTUG, in collaboration with TUG and NTG, are producing a plug-and-play CD-ROM based on Thomas Esser's teTeX distribution. It adheres to the TDS (TeX Directory Structure) tree structure (see, for instance, TUGboat 16(4), pages 401–412).

Presently Unix executables for the following platform/operating system combinations are included: Digital alpha-osf (2.0 and 3.2), Hewlett Packard hpux (9.01 and 10.01), Intel i386 bsd2.0, freebsd (2.0.5 and 2.1.0) netbsd (1.0 and 1.1), Intel i486 (linux and linuxaout), m68k (linux, linuxoldld, and nextstep3), mips (irix 5.2, 5.3 and ultrix4.4) IBM rs6000 (aix3.2 and aix4.1.1) Sparc Solaris (2.4 and 2.5) and Sunos 4.1.3.

As the CD-ROM uses the ISO 9660 standard, the platform-independent files can, in principle be read on all operating systems which are compatible with that format. To preserve the complete Unix/Posix file system information the file tree was recorded with the Rock Ridge extensions, so that long filenames are honoured. Some operating systems,

most notably MS-DOS, do not support these extensions, so that only the ISO 9660 filenames are used, but with this limitation, the files are readable on all systems. At present the complete L^AT_EX Dec. 1995, patch level 2 base and all contributed files are included, plus the most-used font families (CM, DC 1.2, TC, AMS, Euler, psnfss). We plan to publish an updated CD-ROM twice a year (more or less coordinated with new L^AT_EX releases), and we hope to add more packages and fonts in the next issues.

The complete source of the teTeX package is included, which has the source of T_EX, METAFONT, and all the T_EX utilities. Documentation for all of the macro packages on the CD-ROM is included in DVI and PDF format, as well as the original source. A large fraction of the documentation has also been converted to HTML.

The CD-ROM will be officially presented on May 29th in Nanterre (Paris), where GUTenberg organises a seminar on T_EX distributions. Copies will be available from NTG for a price of 35 NLG.

Attending EuroT_EX'95 in Papendal

Michel Goossens

CERN, Geneva, Switzerland
goossens@cern.ch

I arrived with the night train from Basel at 8:29 precisely in Arnhem Railway station on Monday morning September 4th. I decided to take a cab to Papendal, the Sports Center a few km outside of the town, where I arrived just before nine. In no time I was registered and given two keys to my room, where, as expected, I would spend only a minimal amount of time during the four next nights...

Indeed, while walking the fifty meters or so to the breakfast area, I met many known faces, and I had to promise each one of them that we would have a chat later during the conference.

A truly international gathering

EuroT_EX conferences are quite different from the T_EX Users Group conferences, in that many more participants from Russia and Central Europe participate, and this year was no exception. In all 103 T_EX users from 18 different countries (4 from Belgium, 1 from Canada, 7 from the Czech Republic, 2 from France, 17 from Germany, 1 from Hungary, 2 from Lithuania, 18 from the Netherlands, 1 from Norway, 20 from Poland, 15 from Russia, 1 from South Africa, 2 from Spain, 1 from Sweden, 3 from Switzerland, 1 from Turkey, 6 from the United Kingdom, and 1 from the United States of America) came to EuroT_EX'95, where our hosts, the Dutch-speaking T_EX Users Group *NTG*, had prepared a nice, interesting, and extremely dense program of T_EX and text-processing related presentations given by contributors who are active in the various areas mentioned.

After breakfast the members of the Program Committee, Johannes Braams, Chris Rowley and myself, met to make final adjustments to the schedule of the various talks and tutorials, taking into account the availability of the speakers, so that the program could be communicated to the participants of the conference, who now started to arrive.

The EuroT_EX Bus

But before going any further let me explain how such a large group from Russia and Central Europe was able to come to Papendal. It was mainly thanks to the generous contributions of the *NTG*, who donated a sizable sum of money they had collected by selling the 4allT_EX CD-rom(s), plus over \$1800 from the book auction that took place at T_EX Users Group'95 in St. Petersburg Beach (Florida, USA) that a Polish bus was hired. In Warsaw the

Russian, Lithuanian and part of the Polish group boarded, while other (Polish and Czech) participants were picked up later on the way to Papendal. In this way the travel expenses of about forty people were kept to a minimum. This great initiative, which, I hope, will be repeated in the future, shows the enormous solidarity in the T_EX community. It can serve as a proof that T_EX is not only a (or "the") utility to compose beautiful documents in most of the languages written by mankind, but it also brings people of various nations, cultures, and backgrounds together, and thus fosters communication, which is the basis of all human success.

The Conference starts

A few minutes before 2 p.m., the conference was formally opened by Erik Frambach, the President of *NTG*. He welcomed all participants, and then passed things over to me. In my role as President of T_EX Users Group, I congratulated *NTG* for the magnificent work they had done for allowing so many participants from different parts of Europe to take part in the EuroT_EX meeting, and I underlined that T_EX Users Group fully supports all such initiatives and is doing its best to help all T_EX users in the world. I therefore invited everybody who wanted to express constructive ideas about how to develop or improve the actions of T_EX Users Group in the area of supporting the international community better, to find me during the week for a discussion. Finally Johannes Braams, on behalf of the Program Committee, explained the few changes that had to be introduced in the program in order to accommodate a maximum of the wishes of speakers and participants, and he gave a few details on logistics for the rest of the week.

Monday p.m.—Font developments

Thus, well informed, the conference could start with the talk on VFComb, a program for designing virtual fonts by Alexander Berdnikov and Sergey Tirtia from St. Peterburg (Russia). The main aim of their MS-DOS program is to facilitate setting up virtual fonts for use with CM and the Cyrillic LL-fonts. This program is especially interesting in that it constructs font and user-defined ligature tables extracted from various fonts and combines it with metric information from various *tfm*-files. The VFComp program not only supports the full syntax of *.pl* and *.vpl* files, but also adds some symbolic variables and conditional operators to ease the production and debugging of virtual fonts.

The next speaker, Jörg Knappen of Mainz (Germany), described his work on version 1.2 of the DC fonts and the text companion symbol fonts. He first gave an overview of the improvements introduced into version 1.2 with respect to the previous release in order to cope with criticism in the areas of the placement of accents, the design of the quotation marks, plus the outlines for various Polish, Czech, Slovak diacritics, and the height of the umlauts. The *text companion* TC fonts try to address the problem of more clearly separating the symbols present in the CM math fonts from the really mathematical characters, and to make uniformly available to T_EX users several custom signs (currencies, trademarks, some arrows, a musical note, etc.) that are present in the ISO standards 8859-1, 8859-2 and 6937. For practical reasons Jörg proposes to put all those characters into two different fonts, TS1 and TSA. Last but not least, the speaker underlined the importance of design size in fonts, since the linear scaling of a font over a large range gives the wrong results. He therefore proposed encoding the design size of the font in the font name in each case by using the trailing four digits of the name to represent the design in points (multiplied by a factor of 100). He therefore has to rename a few of the present DC font names, since one has only two characters left for weight and shape. For instance the present `dcssi10` would become `dcssi1000`, and `dcbxti10` is `dcbi1000` in the new scheme. Thanks to Jörg's work, and the work done earlier by the L^AT_EX3 team on math font encodings, it is now hoped that a proposal for the two math fonts will be reached soon, so that by next summer the DC fonts will be promoted to the EC fonts, and be accompanied by a useful and standardized set of 256-character math fonts.

Just before the tea break, Jiří Zlatuška from Brno (Czech Republic) gave a talk on how to use METAFONT and T_EX together for typesetting text and graphics.¹ By using T_EX's extended ligature mechanism, label placement on diagrams generated by METAFONT, as well as the generation of curvilinear texts, can be accomplished in one METAFONT pass and only requires a simple T_EX interface. Institutional seals and other logos can thus be conveniently generated.

Multiple languages

After the break Andrey Slepukhin (Sergiev Posad, Russia) introduced the audience to the magnificent world of old Church Slavonic typesetting. Andrey works for the Holy Trinity St. Sergius Lavra publishing house, printers of Bibles and other texts in Church Slavonic. For his project of printing such texts he has to develop high quality fonts, tools for simplifying text input, and put together hyphenation tables. Church Slavonic has a lot of characters and they have very few common parts, so that designing them takes a lot of time. The main problem, however, is that almost every word in Church Slavonic texts has a diacritical mark. Since the placement of the diacritical mark is

not fixed, and the use of the `\accent` primitive has a lot of unwanted side-effects, Andrey developed the various letter-accent combinations as separate glyphs. By carefully studying Church Slavonic texts it was found that some letter-accent combination could only occur in certain locations in a word. In this way, with the help of a set of special macros and using active characters, the number of needed symbols could be kept below 256, the limit of symbols in a single font. Other problems he had to solve were coloring symbols (a practice occurring extremely frequently in such texts), numbering (numbers are represented by letter combinations), and encoding (several different encodings are actively used in the Cyrillic world). To help solve the last problem he proposes using symbolic names to map to character codes and has implemented these ideas in the latest version of his package. He is also working on Type1 variants of his fonts, and on extending his character set to include older Slavonic letters, initial caps, and a special font for headings. Finally, he has expressed a wish that at some time in the not too distant future it might be possible to typeset a multilingual Bible with parallel texts in Church Slavonic, Greek, Latin, Hebrew, and any other language.

Olga Lapko of Mir Publishers (Moscow, Russia) then presented work she has been doing together with Irina Makhovaya and other collaborators of CyrT_EX Users Group, on developing a Russian style for the `babel` system that implements in a user-friendly way the typographics and national peculiarities of the Russian language, such as correct names for some math operators, repeating signs in broken math formulae, and Russian-character enumeration lists. Several font encodings are supported and are interfaced to 256-glyph Russian fonts.

In the next talk Johannes Braams reviewed his work on the `babel` system. The present release 3.5 has a completely rewritten interface to deal with shortcuts, introduces new ways to switch languages, has been made compatible with L^AT_EX2_ε's input and font encoding packages, has added support for a configuration file and many more different languages, and offers an extended syntax for specifying hyphenation patterns.

The final talk of the afternoon was by Petr Sojka from Brno (Czech Republic), who presented his work on hyphenation for compound words. His paper was chosen at TUG'95² as the best technical paper by Knuth for pointing out problems with hyphenation that he himself had not considered. The problem is with long compound words that occur frequently in German, Dutch and Slavic languages. In these languages constituent word parts are not signaled by a hyphen or other fill character, thus making it difficult to find the correct hyphenation in some cases. Petr proposed extensions to the hyphenation algorithm and primitives which would make it possible to deal with these cases. Perhaps an interesting suggestion to be implemented in e-T_EX or Ω?

After dinner Sebastian Rahtz headed a workshop on Acrobat and electronic publishing. He presented tools that

¹He presented this talk also at T_EX Users Group'95, see *TUGboat* **16** (3), p. 223–228, 1995.

²His article is published in *TUGboat* **16** (3), p. 302–305, 1995.

take advantage of \LaTeX to generate hypertext views of a document. In his presentation he made it clear that \LaTeX , HTML, and PDF should be viewed as complementary representations, that each has its advantages and is suited for addressing the specific needs of some applications.

Tuesday a.m.—Graphics and Packages

The Tuesday morning started with a presentation by Andrey Astrelin of Moscow (Russia), who talked about a new implementation of graphics in \TeX . Of the three basic ways of introducing graphics inside \TeX (special fonts, using rules, or exploiting the `\special` command) his implementation uses the approach of extending \TeX 's graphics capabilities by writing graphics commands via `\special` commands into the `.dvi` file, which is then post-processed to realize the required functionality. Each command is represented by a character and zero or more numeric arguments. There are general graphics commands, complemented by path and area commands. The `.dvi` file is processed by a special MS-DOS program, where the graphics `\special` commands are replaced by `emTeX` `\specials` pointing to `.pcx` pictures. These facilities are complemented by macros to place text and graphics on the page and a library of graphics macros that is still under development. It is planned that future releases will offer support for generating `.pk` fonts and PostScript output.

\TeX Plotter, a program to create two- and three-dimensional pictures developed by Alexander Berdnikov and Sergey Turtia of St. Petersburg (Russia), was presented next. The program is written in Pascal for MS-DOS, and plots functions depending on two variables. It allows one to obtain equiline and surface representations of complex functions without running into overflow problems with \TeX 's memory. The program has a menu-driven user interface which allows the creation and viewing of pictures in a variety of forms, including \LaTeX 's `picture` environment, the `epic/eepic` package commands or `emTeX` `\specials`. Future releases will also support `mfpic`, `PCTeX` and EPS files.

Gabriel Valiente Feruglio of Palma de Mallorca (Spain) then gave a detailed overview of the various commutative diagram packages that are available. He compared eleven of them in terms of their ability to generate complex diagrams, quality of documentation and generated output, ease of installation, user interface, resource requirements, and portability. The way a diagram is structured to increase readability is a somewhat subjective matter, often related to points of aesthetics, so that it is difficult to say which package is the best in applications. In his estimation, the familiarity of an author with the syntax of one package is often more important than the intrinsic power of other packages.

After coffee Hans Hagen of Pragma (Zwolle, the Netherlands) presented a paper on typesetting chemical formulae with the `PPCHTeX` system he wrote together with A.F. Otten. The system is based on `PCTeX`. Basic structures for often used chemical structure diagrams are available as macros, and can be combined and “dressed” with radicals

in various way to obtain complex formulae. The syntax is straightforward and logical. It also allows typesetting reaction equations easily and provides for several special features, like fine-tuning the position of radicals, placing small formulae inside running text, and optimizing the depth of subscripts.

The last talk of the morning was by Daniel Taupin (Orsay, France) who introduced `MusiXTeX`, an improved version of his earlier `MusicTeX` system for typesetting polyphonic music with \TeX . `MusiXTeX` was developed together with Werner Icking and Andreas Egler. It uses three passes (`MusicTeX` was a one-pass system), where during the second pass a program `muFlex` is used to compute optimal spacing to position the notes and determine the length for slurs and ties. This makes for more aesthetically pleasing scores. A planned future improvement is cleaner lyrics insertions. Some issues remain between Andreas Egler and the other two developers about future developments and the compatibility between `MusicTeX` and `MusiXTeX`.

Tuesday p.m.—Electronic documents

The theme of the Tuesday afternoon was electronic documents, and Wiegert Tierie of Adobe Systems Benelux (Amsterdam, the Netherlands) started off with a detailed overview of Adobe's Acrobat series of utilities to create, use, store, annotate, send, view and print electronic documents. To do this Adobe developed the “Portable Document Format” (PDF), a language that is based on PostScript and includes commands to implement a hypertext functionality. He introduced the (freely available) Reader—for reading PDF documents—, Exchange—for introducing annotations—, Writer—for directly creating PDF files from applications—, Distiller—for translating PostScript documents into PDF—, Catalog—for creating full-text indexes—, and finally Search—for full-text searches. Via Application Programming Interfaces (API's) developers can easily plug in their own extensions to take full advantage of the capabilities of the Acrobat system, thus allowing an optimal integration of Acrobat into custom products.

Hans Hagen of Pragma (Zwolle, the Netherlands) then told us about his experience using \TeX to generate hypertext documents in PDF. Hans expects that in a few years PDF will be as stable and standard as \TeX is today, so that at that time PDF might well become an important format for distribution of documents. Because PDF is based on PostScript it can combine high typographic quality with hypertext capabilities offered via PDF's `pdfmark` operator. Building a utility able to exploit both \TeX 's and PDF's strong points relies on communicating information from \TeX to PDF via `\special` commands that will contain `pdfmark` instructions. Hans emphasized, however, that writing hypertext documents for viewing on a screen requires a somewhat different approach from optimizing them for paper. He presented a set of points that have to be dealt with, in particular different aspect ratios, a more complex pagebody to optimally place navigation aids, synchronization of screen and printed versions of documents, the

design of the typographic interface, the generation of tables of contents, multiple indexes, cross-references. Other points to be considered are how to handle multiple documents, shared data, status bars, active figures, color, and version control. The speaker thought that TeX was an ideal tool to explore the advantages and limits of electronic documents and that TeX and Acrobat thus formed a nearly perfect match, combining the advantages of both a flexible programming language and a powerful document delivery system.

After the tea break Sebastian Rahtz (Elsevier Science, Oxford, United Kingdom) talked about the experience he had gained working on the `hyperref` package, developed together with Yannis Haralambous of Lille (France). This LaTeX package uses LaTeX's cross-reference commands to transmit the necessary information to PDF and thus provides a rather straightforward and effortless way for LaTeX users to transform their documents into hypertext. Sebastian pointed out some difficulties of this approach, namely the successful handling of fonts, and the generation of back-references from the bibliography and index. Although it thus is easy to generate PDF from LaTeX documents in a way that they are faithful representations of each other, all problems of optimizing the presentation for screen viewing, as explained by Hagen in the previous talk, still remain to be addressed.

During the next half hour or so I gave an introduction to SGML, using HTML as an example of a DTD. I emphasized that using and understanding SGML is straightforward by using examples drawn from HTML. I outlined the structure of a DTD and how it describes a document, its elements, their attributes and the entities that are available. The mathematics and table extensions of HTML3 were presented briefly, but it was emphasized that HTML's main aim is describing the document's structure, not its perfect typesetting, where it is better to consider the PDF solution, so that HTML and PDF are complementary rather than competing technologies.

My second talk presented the LaTeX2HTML tool of Nikos Drakos, which translates LaTeX documents into HTML using a perl program. All standard LaTeX commands are dealt with. Tabular and mathematics constructs are turned into images since they cannot yet be treated by HTML, and the same strategy is applied to user-defined commands or environments. I emphasized that with a little work by the user such user-defined extensions can be dealt with by providing a perl routine for the extension in question, thus increasing substantially the usability of the generated HTML document, decreasing at the same time its size and fragmentation. The hypertext extension package `html` was described and it was shown how it allows the introduction of hypertext elements into a LaTeX document, so that it can be optimized both for printing and hypertext viewing on screen. Finally, work on providing support for the automatic generation of HTML3 and the implementation of HTML3 capabilities in the experimental arena browser were briefly mentioned.

The afternoon was concluded by a panel discussion on electronic documents, chaired by Joachim Schrod of Darmstadt (Germany), with all speakers of the afternoon as panel members. Each of the speakers again emphasized the complementarity of the various approaches to enrich the LaTeX source code with markup that can be used to exploit the hypertext capabilities of the various output media targeted. However, hypertext viewing on a computer screen is quite different from writing books, and therefore care must be taken to design and mark up documents with enough generality to optimize their re-use in the various circumstances. Another interesting question which was raised was the level of control that should be given to the viewer to customize the appearance of the viewed document. Should one allow complete freedom to the receiver to represent the document (font type and size, setting of colors, preferences for lists, page size, etc.) or should the author be able to freeze some or all of the document's appearance (for instance for legal documents). It became soon evident that a general answer to this question was impossible, and that as much flexibility as needed should be given to the receiver. As the hypertext medium, and the freedom to "view as one likes" are still quite new, the plethora of browsers, and possibilities to (over)dress one's documents will soon die out as the technology matures, and some studies in readability and efficiency in communication information via the screen on the Internet come up with some guidelines that will eliminate the more extreme excesses in the area, just as fontitis, or the disease to use as many fonts as possible in one's documents, died out after a few years after people realized that the documents became unreadable.

After dinner a long presentation/discussion session on the achievements of the e-TeX project was organized by the e-TeX team, with Phil Taylor, Jiří Zlatuška, Bernd Raichle, and Friedholm Sowa, who came specially for that day, present. It was an extremely useful discussion, that lasted until midnight, and I am sure that both the members of the e-TeX team and the participants benefited greatly from the exchange of ideas.

Wednesday a.m.—Tools I

The tools section started on Wednesday morning with the presentation by Andries Lenstra (Uitgeest, the Netherlands) of the $\delta\alpha$ TeX package that he developed with his colleagues Seven Kliffen and Ruud Koning. The $\delta\alpha$ TeX system is a series of TeX macros for storing and retrieving data that work both with LaTeX and plain TeX. The system makes it possible to keep source texts short and guarantees data integrity and uniformity in typesetting. Data are stored in $\delta\alpha$ TeX format in a $\delta\alpha T$ file. The format is simple, versatile, easy to learn, and as portable as a TeX document. Conversion utilities to generate $\delta\alpha$ from ASCII files are available. $\delta\alpha$ TeX has sorting-out facilities, supports default fields, conditionals, and wrapping.

Kees van der Laan (Garnwerd, the Netherlands) explained how to use his BLUE's format to manage a database. In

particular, \TeX technical details concerning storage and access of the material are hidden from the user, and since the database code is written in plain \TeX , it can in fact be used with any \TeX format. High-level user commands allows non- \TeX technical users to take full advantage of this tool, which moreover allows one to include pictures and cross-references. Facilities for preparing letters to be sent to multiple addressees are also included.

Philip Taylor (London, United Kingdom) then gave one of his famous pedagogical presentations, explaining how one can use `\csname...\endcsname` constructs advantageously in various often-occurring programming paradigms in \TeX . He showed how this construct helps to clarify the programming logic in many \TeX macros, and thus makes documenting and maintaining such macros more straightforward and more robust. As an interesting side-remark he mentioned the fact that an undefined `\csname-construct` has the value `\relax` can be a disadvantage in some cases. All in all a well-prepared, stimulating, and thought-out talk, for which Phil rightly got the prize for the most pedagogical presentation of the conference. Congratulations Phil!

Just after the coffee break Laurent Siebenmann (Paris, France) introduced his \TeX utility *Occam* that is useful for managing macros. The *Occam* utility eliminates from a set of \TeX macros all those that are not referenced (and thus not necessary), thereby reducing (sometimes significantly) the size of a \TeX source document, at the same time making the maintenance and comprehension of the supporting macro collection easier.

The final talk of the morning was a presentation of *pascal*, a \TeX macro package for typesetting Pascal programs, based on work by Pedro Palao Gostanza and Manuel Núñez García (Madrid, Spain), and presented by the former. This utility allows the user to clearly indicate the structure of a program by using special typesetting conventions. Technically speaking, all reserved words and identifiers become \TeX control sequences, so that no external parser is necessary and layout conventions can be easily enforced. Extensions of these ideas to other structural programming languages such as Modula-2, Ada, or ML should be straightforward.

Wednesday p.m.—General developments in \TeX and \LaTeX

The afternoon was reserved for presentations related to recent general developments in \TeX and \LaTeX . Chris Rowley (Open University, United Kingdom), as a representative of the \LaTeX 3 Team, gave an overview of the activities of the team. Having devoted the Group's resources over the past two years on the development and consolidation of \LaTeX 2 ϵ , the Group will now concentrate on further studies on the way to \LaTeX 3. During his talk Chris also explained the Group's policy on modifying files in the \LaTeX 2 ϵ distribution.

Next Philip Taylor, speaking for the NTS and ϵ - \TeX teams, announced that version 1 of ϵ - \TeX was now available, and gave a list of the extensions which have been introduced. He emphasized that ϵ - \TeX can be used in a 100% compatibility mode with \TeX and he expressed the hope that with time and after a lot of tests, most \TeX users will have enough confidence to use ϵ - \TeX as a valid replacement for \TeX . In fact, when generating the format file with `initex`, one can configure ϵ - \TeX so that it is completely compatible with \TeX , and the extensions and enhancements are disallowed. The new features implemented in this first version are in the areas of additional control over expansions, re-scanning tokens, environmental enquiries, additional marks and debugging facilities, bi-directional typesetting, and a few supplementary primitives.

John Plaice (Laval University, Ste-Foy, Québec, Canada) of the Ω team then gave an update of work done since the summer of 1994 on this 16-bit extension to \TeX . In particular, he described the additional collaboration with a graduate student in Estonia in the area of generalized multi-dimensional typesetting, and recent progress in the area of Arabic. A beta-version now runs on a PC. Dynamic memory allocation is next on the list of features to be implemented in order to reduce the memory requirements of the system when static arrays are used.

It was then Joachim Schrod's turn to introduce the audience to the proposals of the "tds" (\TeX Directory Structure) team, whose members have been working for many months on the definition of an ISO 9660-compliant directory structure that can be used as a plug-and-play run-time system by all operating systems.

A panel discussion on the theme *\TeX quo vadis?* concluded the afternoon. Several members in audience thanked the various teams who are working so hard to improve the functionality of \TeX , \LaTeX , and their user production environments and gave valuable input to the respective development teams, who promised to look at ways of implementing the suggestions put forward.

Wednesday Evening—The Social Event

At 16:30 two buses took all participants to the center of Arnhem, where we had two hours for a walk through the streets of this historic town, perhaps best known from the film "A bridge too far", which described the battle for the bridges over the Rhine in the vicinity of Arnhem. The inner city, which was almost completely devastated in 1944, has since been rebuilt, and has become a modern local center where the importance of the river Rhine is omni-present. It was thus also appropriate that our whole company boarded the ship "MPS Graaf van Bylant", named after a wealthy count who in the 18th century owned large parts in the neighbourhood of Arnhem. It was already getting dark when we left the embankment, and set off for a three-hour tour of the Rhine and the IJssel, taking us along the quays with its wharfs, warehouses, and the apartment building in the background, with their thousands of little lighted windows, each corresponding to a little cell of present-day so-

ciety, people eating, drinking, talking, watching television or just enjoying each other's company: yellow-orange rectangular reflections of human joy and grief, micro-windows on the life of a modern city. But I am sure that these metaphorical considerations were absent from the heads of most EuroT_EXers that evening, since our *NTG* hosts did their utmost best to make the outing as enjoyable as possible. Apart from the excellent food and drinks (including quite a few bottles of Polish and Russian Vodka, contributed by GUST and CyrT_EX Users Group), there were also the MAPS awards for the best papers in the proceedings. They went to Gabriel Feruglio for his overview article of commutative diagrams and Bogusław Jackowski for his beautiful article on the use of EPS and METAFONT (this was, in fact, already his second MAPS award!). On top of that the MAPS editing team decided to award two "special" prizes and it was quite a nice surprise that MAPS chose to honor the "TUGboat Production Team", for their efforts to publish *TUGboat* on time again. With great pleasure Barbara Beeton, Sebastian Rahtz and I myself received the now-traditional MAPS-sweets-cone (a collection of 256 sweets of many different colors, with the necessary "glue" and all other necessary attributes...). And I found it very appropriate that MAPS gave also a special prize to Erik Frambach, who had been doing an enormous amount of visible (and even more invisible) work to make the conference the success it has become. Apart from these formal parts of the evening, there was of course ample occasion for direct person-to-person discussions and other forms of socializing, making contacts, making new acquaintances and friends, and around midnight, when we arrived back at the Papendal Sport Center, we could all go to bed trying to assimilate all those new and enriching impressions and contacts of this memorable Wednesday Evening EuroT_EX'95 cruise.

Thursday a.m.—Tools II

The last formal session of the conference on the Thursday morning started with a talk by Philip Taylor who tried to convince the audience that T_EX is quite an unsuitable language for marking up documents. He came to this conclusion after he had to deal with the production of a book in the field of linguistics, and had to communicate with an author who did not know T_EX. Phil therefore developed a syntax, <ATML>, for "A T_EX Markup Language", where all markup is enclosed between the triangular brackets < and >, thus disallowing all direct T_EX markup. Phil explained the precautions that he had to take to implement this scheme, and discussed the advantages and drawbacks. It seemed clear that for non-T_EX-aware authors this approach can certainly maximize the efficiency for preparing documents where T_EX should be used as the typesetting engine, but where otherwise consistency and structural markup can be checked at a level of <ATML>.

Similar ideas were supported by Antonin Strejc (Prague, Czech Republic) who described the W95 environment, where contributors to a conference can use an MS-DOS based authoring system, which is a menu-driven interface

to L^AT_EX. The authors can specify the information needed for preparing their article (title, author, affiliation, abstract, etc.) via these menus, and also have the possibility of entering other text elements, like lists, equations, tables, and figures. Thanks to this approach, which guarantees consistent and correct markup, the editors of the proceedings were able to typeset almost 1000 pages by over 430 contributors in about two weeks. The speaker emphasized that such a system needs close cooperation between the organizing committee and the typesetter, and it presupposes that the instructions have been announced to the contributors about six months before the conference. To minimize the need for direct hotline-type support (it was found that with the W95 system only 1–2% of the authors needed personal help) a sufficiently self-documenting help facility should be provided.

Various ways of improving the archiving of scientific documents by optimizing the ways external material can be included by standardized keywords for T_EX's `\special` commands were discussed by Laurent Siebenmann, the next speaker. Although his ideas were quite interesting, it was not completely evident that his demands were not already dealt with by the work of the `dvi`-standard committee, that had a few meetings at T_EX Users Group'95 in St. Petersburg, and Laurent was invited to contact the members of that working group. The speaker also said a few words about the effort concerning atomic fonts...

After the coffee break Kees van der Laan showed how one can implement indexing in a one-pass T_EX run. Although only moderate indexes can be dealt with in this way, the approach is nevertheless quite useful for smaller documents, or for proofing indexes on a chapter-by-chapter basis. For the markup Kees followed closely the circumflex (^) notation proposed by Knuth in the T_EXbook. Sorting is possible and several methods of ordering (lowercase/uppercase, accented/non-accented/word ordering) are available. This indexing system is part of the BLUeT_EX system.

Stanislav Brabec (Prague, Czech Republic) discussed his typesetting system, based on plain T_EX. He uses T_EX's powerful programming facilities to generate macros for managing references, contents tables, defining page layout, with a flexible facility for adding cropmarks, specifying margins, preparation of booklets, and typesetting in landscape mode. His systems `upages.tex` allows one to interpret the input stream on a token-by-token basis, is able to prepare output for PostScript devices, with support for rotation and scaling; it generates device-independent color, with provision for color signatures and separations, and has primitives for line-drawings.

The last formal presentation of the conference was by Bogusław Jackowski (BOP, Gdańsk, Poland), who presented his METAFONT-EPS interface. The heart of this interface is the `MFTOEPS METAFONT` package which provides the necessary definitions for translating the descriptions of graphics objects from METAFONT to PostScript. The PostScript code is written to a log file, that can be post-processed to generate EPS files that can be read by popu-

lar PostScript drawing tools like Adobe Illustrator, Coral-Draw, or Fontographer. The package comes with two such utilities, one written in awk, the other, more general, but slower, in \TeX . This system is thus merely useful for generating PostScript code that should be edited further by the tools mentioned above. For generating end-user PostScript, John Hobby's METAPOST program is probably more suited. The author is also working on a program PSTOMF, that would translate EPS files (via the `ghostview` program) into METAFONT, thus completing the link between METAFONT and PostScript and allowing everybody to make use of the advantages of both languages.

It was at about half past twelve when Johannes Braams, in the name of the Euro \TeX '95 Program Committee, thanked all speakers for their contributions, and Wietse Dol for his work on preparing the Proceedings.³ He then invited Andrey Slepukhin to step forward to receive the prize for "Best Presentation of Euro \TeX '95", not only for his seminal work in the area of font creation of beautiful Church Slavonic letters, but also for developing solutions for the many technical problems relating to typesetting complex documents. A prize for the most pedagogical presentations (not only at this conference, but also on other occasions) was given to Philip Taylor. Sincere congratulations to both recipients of these prizes.

I then, in the name of the international \TeX Organization, \TeX Users Group, thanked the \mathcal{NTG} organizers for their dedication and hard work in making Euro \TeX '95 possible. I also thanked the many contributors, and last but not least the participants to the conference, and I invited everybody to the joint \TeX Users Group-Euro \TeX meeting that will take place next year 1996 in Dubna (Russia) from Sunday July 28th to Thursday August 2nd, 1996. This will be the first time a conference will take place in a country whose major language does not use the Latin alphabet. A unique occasion to remain up-to-date on what is happening in the world of \TeX and friends, and to become acquainted with the rich Russian culture thanks to direct contact with Russian people and visits to famous monuments.

It was \mathcal{NTG} 's President, Erik Frambach, who had the honor of having the last word. He gave a complete list of

the people who had spent many months since the beginning of 1995 to make the conference a success: yes it takes many people to do all the little (and not-so-little) tasks to have everything ready on time. Then he announced that the first 4all \TeX prize had been awarded to Eberhard Mattes, the father of *em* \TeX , the \TeX engine that lies without doubt at the heart of most \TeX installations in the world. A cheque of 3, 141.59 German Marks will be sent to Eberhard, and the applause that accompanied the announcement underlined how much all \TeX users worldwide, especially those using PC's, appreciate his extremely valuable and continuous contributions. With this Erik formally closed the conference in a grandiose way, and hoped to see everybody again next year on the Volga in Dubna.

Thursday p.m. and Friday a.m.—Tutorials

The Thursday afternoon two tutorials ran in parallel, namely one by Bogusław Jackowski on METAFONT and another by Piet van Oostrum on page layout with L \TeX . Both teachers gracefully agreed to repeat their respective tutorials the next morning to give a maximum number of participants the chance to attend both and to give those who had to leave already the Thursday evening the chance to attend the one of their choice. As I had meetings scheduled with various people during the time of the tutorials I was unable to go to either of them, but listening to those fortunate enough to have attended I learned that they were both an enormous success. Congratulations to both Bogusław and Piet for their careful preparations and pedagogical presentations.

Finally, on the Friday afternoon, those interested were able to join our friends from Central and Eastern Europe in a visit of Amsterdam with the Euro \TeX bus, a second (and especially appreciated) "social event".

The Saturday morning with the departure of the Euro \TeX bus for Warsaw via Berlin, the last Euro \TeX 'ies left the Papendal Sports Center. Thus Euro \TeX '95 became history and will henceforth be remembered as the latest (in fact the ninth) in the series of European \TeX Conferences. Thank you \mathcal{NTG} , thank you Erik, and Wietse, and Gerard, and all your colleagues for that week of intense \TeX happiness, and see you all next July in Russia!

³Copies of the Euro \TeX '95 Proceedings—counting more than 440 pages—are available at the price of 50 Dutch Guilders (postage included) from NTG, P.O. Box 394, NL-1740 AJ Schagen, The Netherlands.

BachoT_EX 1996

Wietse Dol en Erik Frambach

De herinneringen aan EuroT_EX 1994 zijn nog steeds bijzonder plezierig. Niet alleen omdat het programma vol stond met goede sprekers (en bekende namen uit de T_EX-wereld) maar ook de vriendelijke en gemoedelijke sfeer in Gdąnsk. Volgens Kees van der Laan en sinds vorig jaar ook Erik Frambach is de sfeer in Polen altijd fantastisch en zijn de mensen bijzonder aardig, hulpvaardig etc.

Natuurlijk hebben we allemaal het reisverslag (BachoT_EX 1995) van Erik gelezen. Helaas moeten wij zijn superlatieven met een korreltje zout nemen. Immers heeft hij een Poolse vriendin en ziet hij derhalve in de lentezon alles met een gekleurde bril.

Toen in Arnhem (EuroT_EX 1995) de Poolse delegatie ons uitnodigde voor BachoT_EX 1996 werd al snel besloten om daar heen te gaan. Kees, Erik en Wietse maakten de afspraak om per auto naar Polen te vertrekken. Op maandag 29 april om 4.15 uur stond Kees bij Erik op de stoep en vertrok het gezelschap richting Duitsland. De reis verliep bijzonder voorspoedig. Na 8 uur reizen stonden we reeds aan de Poolse grens. Van Phil Taylor hadden we een reisbeschrijving gekregen en het advies om een bepaalde grensovergang te nemen. Dit om uren wachten aan de grens te voorkomen. Inderdaad Phil bleek betrouwbaar te zijn. Binnen 5 minuten stonden we in Polen. Bijzonder groot was ons leedvermaak toen we woensdag avond van Volker Schaa (Dante vertegenwoordiger) hoorden dat hij twee uur voor de grens had moeten wachten.

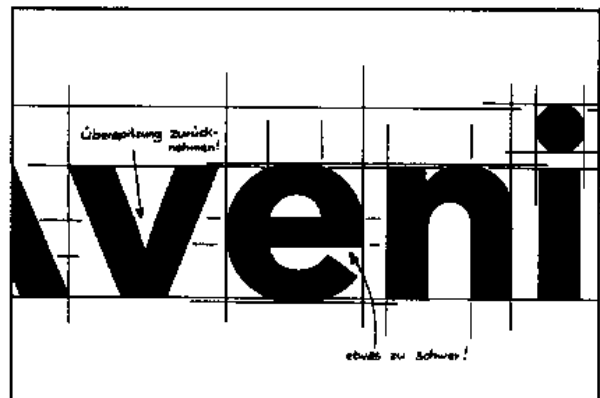
Toen we om 19.00 's avonds (na 1100 km) aankwamen moesten we met handen en voeten (en Kees zijn woordenschat aan Russisch) de beheerder van het vakantieparadijs duidelijk maken dat we ons vakantiehuisje wilden betrekken. Helaas bleek dat we volgens zijn boekingen een dag te vroeg waren aangekomen. Hier bleek al hoe gemoedelijk alles in Polen gaat: binnen no time hadden we de lakens voor onze bedden en werden we naar een hutje gebracht. Op ons verzoek ging de bar open voor een drankje. Vermoeid konden we gaan slapen om de volgende ochtend naar Poznań te gaan om Julita Boland op te halen.



Het Poolse openluchtmeuseum (een van de vele).

Op dinsdag ochtend (7.30 uur) zijn we vertrokken naar Poznań en hebben daar de stad bekeken. Daarna via een toeristische route naar Toruń (de stad van Copernicus) genomen. Julita bleek een voortreffelijke gids. Ze wist ons langs een soort Openlucht Museum (à la Orvelte) te leiden waar we de bouwwerken uit het Poolse verleden konden bekijken. 's Avonds voor bijna geen geld voortreffelijk gegeten in Toruń. Polen is een bijzonder goedkoop land: benzine voor nog geen gulden per liter en een goed avondmaal voor 15 gulden.

Op woensdag zijn we richting Gdańsk vertrokken om daar in de buurt het slot Malbork te bezoeken. Een gigantisch slot met een rijk verleden. Net na de tweede wereldoorlog lag het bijna geheel in puin en is door de Polen bijna volledig steen voor steen hersteld. Op woensdag avond kwamen de meeste deelnemers voor de conferentie binnen en werden de eerste T_EX-contacten gelegd/hernieuwd.



Andrzej Tomaszewski liet ons weer typografische verfijningen zien.

In de kofferbak van Kees' auto lag een grote stapel CD's en EuroT_EX proceedings. Meteen op woensdag avond konden de eerste exemplaren al van eigenaar verwisselen. Het aantal deelnemers aan BachoT_EX was ongeveer 80 mensen en in de drie dagen van het congres werden er 17 CD's verkocht. Een opmerkelijke hoeveelheid. Op woensdag avond kregen wij te horen dat Erik en Wietse op donderdag avond een uur de tijd kregen om een verhaal over 4T_EX en Windows te houden.

Donderdag de eerste dag van de conferentie. Helaas voor ons bleken de meeste voordrachten in het Pools te zijn. Gelukkig kon Julita (net als vorig jaar) als vertaler optreden. Dit jaar was de groep voor vertaling gegroeid: 3 Nederlanders, 1 Duitser en 1 Hongaarse. Veel hebben we niet opgestoken van de Poolse verhalen (af en toe wat vlagen en

demo's die beter te volgen waren). Na een goed avondmaal konden Erik en Wietse op hun gebruikelijke wijze interactief 4T_EX demonstreren. Na de demonstratie 4T_EX voor Windows werden alle deelnemers uitgenodigde voor een "Dutch night" in onze hut. Kees had namens ons drieën een heleboel Nederlandse lekkernijen meegenomen (berenburger, oranje bitter, advocaat en wijn, tig soorten kaas, gember, sardines, haring, toastjes, drop en nog meer). Het was dan ook niet verwonderlijk dat pas om 3.00 's ochtends de meeste mensen maar eens hun hut opzochten. Tijdens de gezellige avond vroeg Wietse zich af wat het zou kosten om een GUST lidmaatschap voor het leven te verkrijgen. Na wat onderhandelingen besloten de drie Nederlanders en Volker Schaa meteen om lid voor het leven te worden (geen geld voor ons maar hopenlijk voor GUST een enorme financiële injectie om leuke/nuttige dingen te doen).

	Avia	Baltic	Calisia
Avia	0:0	1:5	0:1
Baltic	0:0	1:5	7:21
Calisia	3:2	13:21	0:1

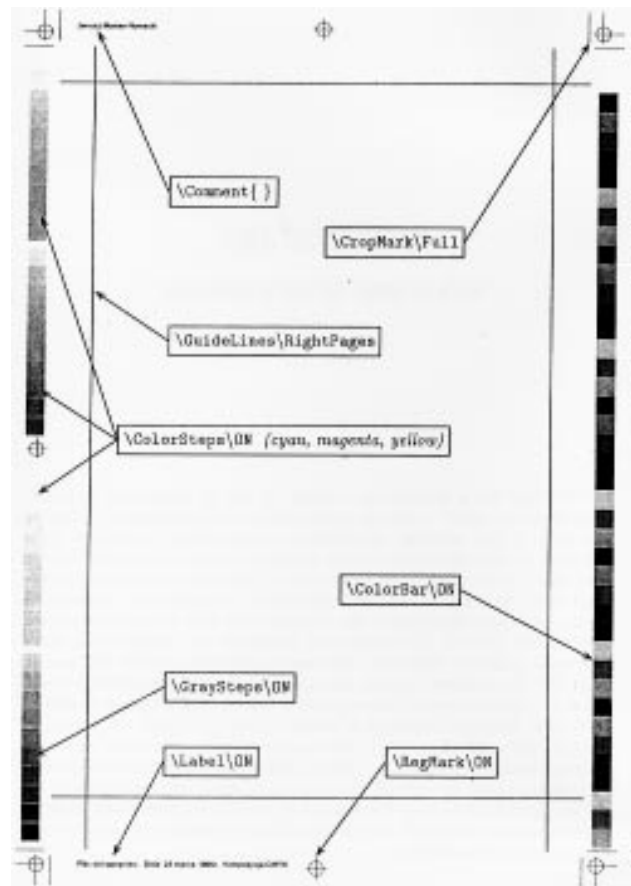
Bogusław Jackowski toonde hoe T_EX en PostScript kunnen samenwerken in tabellen.

Vrijdag was de dag waarop Kees zijn verhalen mocht houden. Het eerste verhaal ging over Metafont en Metapost (zie ook deze MAPS), het tweede verhaal ging over turtle graphics (zie volgende MAPS). Wederom een mooi visitekaartje voor de NTG. De avond was de avond van het grote kampvuur. Niets is gezelliger dan met een meute mensen een enorm fikkie stoken en dan er met zijn allen worstjes te grillen. Helaas begon het om 23.30 uur te regenen waarna we een hut opzochten om verder feest te vieren. De "Polish night" bestond uit veel sterke drank, prachtige gitaar muziek en gezang. Geen wonder dat Kees al jaren verliefd over Polen praat.

Zaterdag de laatste dag ging bijzonder snel voorbij (lag dat aan die kater of was er nog iets anders?). Na de ochtend met lezingen was het tijd om van de meeste mensen afscheid te nemen. Ook Julita moest weg, waarna de Dutch boys weer onder elkaar waren. Na een middag van "netwerken" besloten we zaterdag om 19.30 uur naar huis te rijden. De terugreis ging voorspoediger dan de heenreis en om 7.00 uur zondag ochtend waren we weer in Groningen.

De GUST-meeting was bijzonder gezellig en wij kunnen dan ook iedereen aanraden om eens naar Polen te gaan. Wellicht is het een goed idee om over twee jaar de tweede NTG-lustrum tezamen met de eerste GUST-lustrum te vie-

ren. Polen en Nederlanders liggen elkaar wel en dat is de eerste stap in een succesvol lustrum. In Polen zitten een aantal zeer goede T_EX, Metafont en Postscript gurus. Elke keer ontdekken we weer nieuwe utilities, trucjes en handigheidjes die men in Polen al tijden gebruikt. Polen is (net als Rusland) een plain T_EX-land. De geplande cursus L^AT_EX kon wegens gebrek aan belangstelling (drie inschrijvingen) niet doorgaan. Veel dingen die voor ons gewoon zijn (CTAN, FTP en CD's) zijn voor de meeste Polen niet toegankelijk. Dit was duidelijk merkbaar op BachoT_EX waar velen zeer veel floppen kopieerden vanaf een PC (de eigen Poolse CTAN-opzet). Ook zie je binnen de Poolse T_EX-wereld veel mensen op de zelfde manier werken: batch files en de Volkov commander (een Norton commander). Freeware software is voor hun de oplossing om niet alleen gratis een T_EX-shell te hebben maar door hun handigheid en inventiviteit heeft iedereen een gebruikersvriendelijk en krachtig systeem. Niet iedereen heeft 4T_EX nodig!



Janusz Nowacki liet zijn macro's zien om professioneel drukwerk te maken.

The 17th Annual T_EX Users Group Meeting

ΠΟΛΥΤ_EX

July 28 – August 2, 1996
TUG, CyrTUG, JINR

Time has arrived when T_EX, the Polyglot, who already for many years happily “spoke” a lot of different languages written in the Latin alphabet, starts extending its knowledge also in the field of other alphabets. During the Summer of 1996 a visit is planned to Russia where T_EX will have its first practical session in Cyrillic.

Russia is that large country, with its enormous planes, inhabited by those enigmatic Russians, who started the century with a Revolution, and ended it with “Perestroyka”. The country who founded a brilliant mathematical school, colonized the Cosmos, conquered the world with its literature, music, and ballet.

So what awaits the T_EX user who decided to attend TUG’96?

At Moscow’s international Airport Sheremetyevo-2 the participants to the conference will be met by a member of the organizing committee and be taken by bus to Dubna, the town where from July 28th to August 2nd the 17th TUG conference will take place (see below for more about Dubna). If, for some reason or another, you see cold, the taiga, and white bears appearing, then we can reassure you, Dubna is in the European part of Russia, and if you want to see the ice on the nordic Oceans, you will have to travel further than you would to reach the subtropics of the Black Sea shore. Indeed, you will soon find that the pine forest in which the comfortable town of Dubna is situated will remind you of a park. AT the end of July, the weather is mostly warm and sunny, with an average temperature of +28C°. The guests will be housed in a comfortable hotel on the banks of the Volga river in single or double rooms (hot water, shower, telephone, television). The Russian “cuisine” is characterized by its abundance, so one can forget about slimming.

The social-cultural program includes a picnic on the picturesque banks of the Volga, where we will be taken by boat, a bus excursion to Sergiev Posad (the center of the Russian Orthodox Church, where Andrey Slepkhin works—see his article on page....), and, the last day a visit to Moscow, where, at the end of the day, the participants can be dropped of at the airport to fly home, or at one of the railway stations, if they want to prolong their visit.

The conference’s scientific program will be announced as soon as we receive from you, dear readers, proposals for presentations, courses that you would like to teach or attend,

poster sessions, or any problem or subject that interests you. Please send you suggestions to the conference electronic address TUG96@pds.jinr.ru.

Write to us—all your proposals will enrich the theme our theme:

$$\text{ΠΟΛΥΤ}_{\text{E}}\text{X} = \left\{ \begin{array}{l} \text{Polytechnic} \\ \text{Polymath} \\ \text{Polyglot} \end{array} \right.$$

Practical Information

Conference cost

The Conference Committee foresees a cost in the range 550–600 USD. This sum includes the complete cost of the conference, namely the registration fee, lodging (6 nights with six breakfasts, lunches, and dinners), coffee/tea breaks, social events, and transport from Sheremetevo Aeroport to Dubna. The payment should be made in the following way: a *non-refundable* sum of \$100 per person should be transferred to the following Moscow bank account *before June 1st*:

Name of Bank	Credit-Moscow Russia, Moscow, 113054 6-TH Monetchikovsky Per., 8
Swift Code:	CRMORUMM
A/C N:	00107091799

Upon receipt of that sum an official invitation, necessary for obtaining a visa will be faxed to the participant—so be sure to include your FAX number. The remainder of the registration fee (between \$450 and \$500) will be payable *in cash* upon arrival at the Conference site in Dubna (no credit cards or cheques can be used in Dubna). For participants of economically weaker countries or for students we hope to arrange partial support via sponsors.

Visas

Most of the visitors from outside Russia will need a visa to attend the Conference. Therefore, for arranging a visa into Russia, participants should inform the Conference Secretariat (Mrs. N. Dokalenko, E-mail nataly@ypr.jinr.dubna.su; Fax 7 095 975 2381

or 7 09621 65 891) of their and (possibly) the accompanying person(s)'s full name, date of birth, citizenship, passport number, arrival and departure dates. The Secretariat will forward by fax the visa support message to the participants with which they should apply for visas to the nearest Russian Embassy or Consulate. Please note that you should apply for a visa valid for Dubna, Moscow, and Sergiev Posad¹.

Transportation

The Organizing Committee will arrange direct transportation by bus from Sheremetyevo-2 airport to Dubna (130 km north of Moscow). Not later than four working days before a participant wishes to be met at the airport should the Secretariat be informed of the flight number, precise date and time of arrival (Moscow time). It is our intention to have each participant met by a member of the Organizing Committee, who will accompany the Conference guests to Dubna.

Organizing Committee

Koren'kov Vladimir Vasil'evich (Dubna) – chairman
korenkov@cv.jinr.ru

Maxovaja Irina Anatol'evna (Moscow) – chairman
irina@mir.msk.su

Dokalenko Natal'ja Mixajlovna (Dubna)
nataly@ypr.jinr.dubna.su

Luk'janov Stanislav Olegovich (Dubna)
lukyanov@jinr.dubna.su

Kuznetsova Marina Vilorovna (Moscow)
cyrtug@mir.msk.su

Korobova Galina Aleksandrovna (Dubna)
korobova@cv.jinr.ru

Strelkov Sergej (Moscow)
Aleksandrovich

Rahtz Sebastian (Oxford)
s.rahtz@elsevier.co.uk

Program Committee

Goossens Michel (Geneva) – chairman
goossens@cern.ch

Pankrat'ev Evgenij Vasil'evich (Moscow) – chairman
pankrat@shade.msu.ru

Burbank Mimi (USA)
mimi@scri.fsu.edu

ZHabitskaja Tat'jana Jakovlevna (Dubna)
publish@pds.jinr.ru

Hohlov Jurij Evgen'evich (Moscow)

Welcome to Dubna!

Dubna was founded in 1956 when the Convention establishing the Joint Institute for Nuclear Research was signed. The town is situated on the picturesque banks of the Volga

river and the Moscow sea 120 km to the north of Moscow. One can reach Dubna from Moscow for 2 hours going by car, by bus or by express train. It will take you 1.5 hours to go to Dubna from the Sheremetyevo-2 international airport. Water ways connect Dubna not only to the Russian Volga cities, but also to the waters of the Black, the Caspian, the Baltic and the White seas.

There is no harmful environmental impact of the plants, this together with the large tracks of forest in the environs of Dubna, vast water area with small islands is quite favorable for the sphere of tourism and rest. The Volga embankment is one of the prettiest parts of the town. At springtime the streets of Dubna are full of lilac odour, the apple-trees are pink-white; in summer lime-trees, maples, birch-trees and poplars make the town seem totally green; in autumn the town is all golden excepting the ever-green old pine-trees. The town's modern look harmonizes with the quietness of the surrounding forest. The town was built in the midst of a forest. There are separate patches of trees in the town itself, and the town park is just a part of the forest. It takes just a few minutes to get to the forest from the shopping centre on foot. A few minute's walk and you are outside the city limits!

Small as it is, Dubna is a real metropolis. It is a scientific metropolis. It is a *big little city*, as a visiting american scientist called it many years ago. Since the foundation of the Joint Institute for Nuclear Research (JINR) the name of Dubna has constantly been in the pages of the world's newspapers and journals. Dubna is one of the world centres for fundamental research in nuclear physics. The Joint Institute plays an important role as a coordinator of the investigations of the scientists from 18 JINR member-state institutes. The wide international scientific and technical cooperation is one of the fundamental concepts of the JINR.

Dubna is indeed a town of international friendship. Foreign speech can be heard everywhere. But the words, no matter in which language they are pronounced are clear to everybody: friendly cooperation and fraternity unite all the physicists and mathematicians living and working in Dubna, into an international scientific community.

On October 3, 1994 Dubna opened its doors to the first university, "International University of Dubna: Nature, Society and Man." The university is composed of five "cathedra" of chairs, including socio-economic sciences, ecology and earth science, computer education, linguistics, and health and physical education. Two more cathedras - law and government and technology - are also being contemplated.

The town has a great experience in holding international conferences, exchange of delegations between countries in the sphere of science, education and culture. Dubna and La Crosse (Wisconsin, USA) are sister cities. Dubna is famous for its hospitality. Famous scientists, public figures and statemen from different countries visit Dubna. They

¹Those planning to visit other cities in Russia should obtain themselves the relevant documents and join them to the visa application, so that the needed names of the places to be visited can be entered on the visa form.

are always impressed by the gracious welcome they receive in Dubna and discontinual generosity which Dubna residents demonstrate.

A few words on Moscow

Moscow, the capital of Russia, has a population of some nine million people. It is a city rich in cultural, architectural and historical monuments, and, at the same time, a rapidly developing modern urban community having brand-new blocks of flats, long, straight and broad avenues, parks, gardens, stadiums, schools, cinemas, department stores, recreation centres, bridges and highways. Trough forward-looking, it cherishes the memory of its past, and its old sections lend it a special charm.

Preliminary Program

– Saturday p.m. –

Arrival and registration
Dinner

– Sunday a.m. –

Late Registrations
Opening Ceremony

– Sunday p.m. –

Fonts

KAREL PÍŠKA “Cyrillic Alphabets”
JÖRG KNAPPEN “DC and TC Fonts—A Status Report”
REI FUKUI (Presented by KNAPPEN) “TIPA: A New IPA Font for L^AT_EX”
OLGA G. LAPKO “Full Cyrillic: How Many Languages?”
A.S. BERDNIKOV and O.A. GRINEVA “Computer Modern Typefaces as the Multiple Master Fonts”
A.S. BERDNIKOV and S.B. TURTIA “VFComb 1.3—A Program to Simplify Virtual Font Management”

– Monday a.m. –

Encodings and Multilingual Support I

YANNIS HARALAMBOUS and JOHN PLAICE “Omega and Cyrillic Alphabet Languages: a Synthesis of Available Resources”
RICHARD J. KINCH “Extending T_EX for Unicode”
V.A. KONYSHEV, L.N. ZNAMENSKAYA, and S.V. ZNAMENSKII “Various Russian Encoding Problems and a New Cyrillic Font Set”
PETER A. OVCHENKOV “Cyrillic T_EX Files: Interplatform Portability”
ALEXANDER I. ROZHENKO “On Cyrillic T_EX and Multilingual Support”

– Monday p.m. –

T_EX Systems I

JOHN PLAICE and YANNIS HARALAMBOUS “Latest Developments in Omega”

DAG LANGMYHR “StarT_EX”

GABRIEL VALIENTE FERUGLIO “Do Journals Honour L^AT_EX Submissions?”

S.V. ZNAMENSKII and D.E. LEINARTAS “A New Approach to T_EX-related Programs with a User-friendly Interface”

17:00–18:30 TUG Business Meeting

– Tuesday a.m. –

T_EX Systems II

IRINA MAKHOVAYA “T_EX in Russia: ab ovo”

JOSEPH V. ROMANOVSKY and FEDOR L. YELTSOV “Strategies of L^AT_EX Insertions”

IVAN G. VSESVETSKY “The Straight Gate to T_EX”

LAURENT SIEBENMANN “Electronic Posting and Archiving of T_EX-composed Scientific Documents”

STRELKOV S.A. AND G.R. EPSHTEIN “Making Indexes for VINITI’s ‘Mathematic’ Abstracts Journal”

M. I. GRINCHUK “How to Teach T_EX to Repeat Signs when Breaking Formulae”

– Tuesday p.m. –

Buses depart to Sergiev Posad (packed lunch)

– Wednesday a.m. –

Graphics

KEES VAN DER LAAN “Turtle Graphics and T_EX—a Child Can Do It”
KEES VAN DER LAAN “Graphics and T_EX—a Reappraisal of METAFONT/MetaPost”
A.V. ASTRELIN “Graphic Algorithms: a Common-use Library”
A.S. BERDNIKOV, O.A. GRINEVA, and S.B. TURTIA “Some Useful Macros to Extend L^AT_EX’s picture Environment”

– Wednesday p.m. –

Workshop on Electronic Documents

MICHEL GOOSSENS “L^AT_EX to/from HTML—an Update”
SEBASTIAN RAHTZ “L^AT_EX, Hypertext and Acrobat”
MICHEL GOOSSENS and SEBASTIAN RAHTZ “Overview of the Latest Developments in Electronic Publishing: HTML/SGML, DSSSL, Style Sheets, Amber, Java, ...”

JOACHIM SCHROD “The TDS File System—A Short Introduction”

– Thursday a.m. –

Encodings and Multilingual Support II

M.M. VINOGRADOV “A User-friendly Multi-function T_EX-interface Based on MultiEdit”

YU. E. HOHLOV, S.V. KLIMENKO, E.V. PANKRATIEV, and S.V. ZNAMENSKII “The New ‘Russian T_EX’ Project”

OLGA LAPKO and IRINA MAKHOVAYA “Questions of Babel and Cyrillic Compatibility”

A. SLEPUHIN “Methods of Multilingual Text Processing”

YANNIS HARALAMBOUS “Extending Al-Amal to Unicode”

12:30 Formal Closing Ceremony of TUG ’96

13:00 We will adjourn for an excursion on the Volga, with a farewell party

– Friday –

All Day Excursion to Moscow (packed lunch)

At the end of the day those not returning to Dubna to participate in courses can be left in the centre of Moscow or at the Airport.

– Saturday/Sunday –

To Be Announced

Courses or workshops for those interested.

De kunst van het programmeren

Dirk van Delft

NRC Handelsblad*
nrc@nrc.nl
http://www.nrc.nl/

28 maart 1996

Abstract

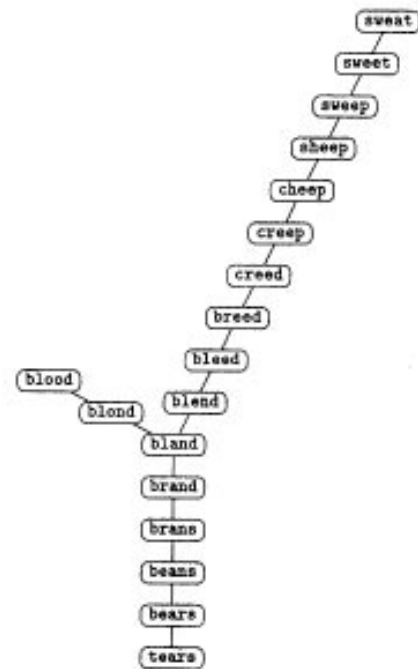
Sinds 1962 werkt Donald Knuth aan een informaticabijbel die over twintig jaar af moet zijn. Tussendoor ontwierp hij een tekstverwerker met eigen typografie en legde hij een Mona Lisa uit dominosteentjes.

In 1952 meldde Donald Knuth, toen veertien jaar oud, zich twee weken schoolziek, zogenaamd met maagpijn. In werkelijkheid ploegde de aanstaande informatica-pionier zich stiekem op zijn kamertje in Milwaukee door een woordenboek, op zoek naar woorden die zich uit de letters van 'Ziegler's Giant Bar' lieten samenstellen. De lokale fabrikant had deze wedstrijd uitgeschreven. Donald kwam op een totaal van 4.500, tweeduizend meer dan de jury, en dat terwijl hij vergeten was de apostrof te benutten. Toen zijn ouders de gedrevenheid van hun zoon opmerkten, hielpen ze de lijst uittikken en het resultaat was dat de hele klas een reuzenreep kreeg, en de school een tv-toestel.

"Ik hou van taal", zegt Knuth, op bezoek in Amsterdam ter ere van het 50-jarig bestaan van het CWI, het Centrum voor Wiskunde en Informatica. "De eerste keer dat ik de krant haalde was als vierjarige boekenwurm, opgesloten in de bibliotheek omdat ik de tijd was vergeten. Als kind schreef ik radioshow's en op mijn twaalfde zette ik thuis een schoolkrant in elkaar, met zelfbedachte puzzels. Woorden hebben zowel betekenis als vorm, en juist de combinatie geeft me plezier. Zo heb ik twintig jaar lang vijf-letterwoorden verzameld — bij 5.757 ben ik gestopt, eigennamen als Knuth liet ik achterwege — en noteerde ik wanneer en waar romanpersonages elkaar in *David Copperfield* of *Anna Karenina* troffen. In mijn informaticaboeken, waar ik duffe voorbeelden per se wil vermijden, maak ik van die lijsten dankbaar gebruik."

Zoals in de *Stanford GraphBase*, een verzameling 'leesbare' computerprogramma's en datasets die Knuth als hoogleraar informatica aan Stanford University (Californië) in 1994 publiceerde — uitgeroepen tot het beste informaticaboek van dat jaar. Een 'graaf' is in de wiskunde een verzameling 'knooppunten' die op zekere manier met elkaar in verbinding staan: de afstandstabel van de Europese hoofdsteden, of een sociogram van de ministerraad. Knuth: "Grafen zijn speelgoed voor de informaticus. Die zoekt in mijn corpus aan vijfletterwoorden de kortste weg van *words* naar *graph* als steeds één letter anders mag. Het

gaat dan om het optimum bij een zeer groot aantal mogelijkheden. Om dat snel te vinden heb je efficiënte algoritmes nodig, zoals dat van Dijkstra uit 1959. Dat leidt op een hedendaagse computer in een vloek en een zucht tot de serie *words-wolds-golds-goads-grads-grade-grape-graph*. Overigens lukt het niet altijd, waar *bares* en *cores* in de 'buren' zwemmen — beiden 25 — hebben *ocean*, *below* of *music* er geen een."

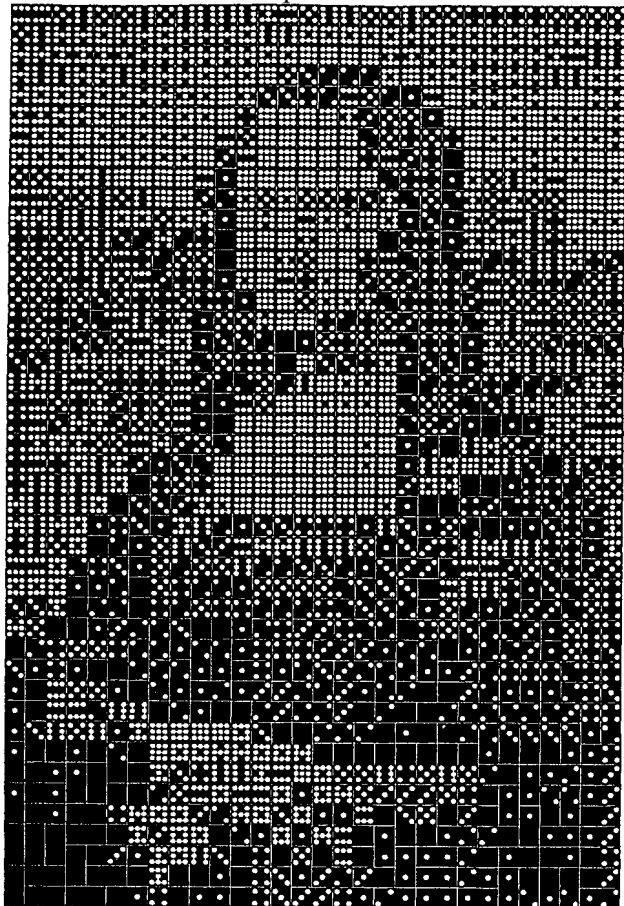


Figuur 1: De dataset 'words.dat' uit de *Stanford GraphBase* bevat 5.757 vijf-letter-woorden. Het programma 'ladder' zoekt 'buren' die in één letter verschillen. Het voorbeeld toont hoe de woorden *blood*, *sweat* & *tears* (naar een popgroep uit de jaren zestig) in zo min mogelijk stappen in elkaar overgaan.

Knuth heeft met zijn *Stanford GraphBase* een standaard willen zetten. "Een algoritme kan altijd beter, één bril-

*Bijlage 'Wetenschap & Onderwijs' blz. 3. Het artikel is in elektronische vorm via docu@nrc.nl bij het NRC Handelsblad verkregen. Copyright ©NRC Handelsblad.

jant idee en het programma loopt duizend keer sneller. Maar rekentijd zegt weinig als computers verbeteren en uitgangsgegevens per gebruiker verschillen. Mijn Stanford GraphBase ondervangt dit door standaard-datasets en -programma's te bieden die iedereen mag gebruiken, zolang hij er maar niet in knoeit. Eindelijk beschikt Tibet over dezelfde digitale Mona Lisa als Californië en kunnen algoritmes tenminste eerlijk worden vergeleken. Daarmee is de weg is vrij voor experimentele computerwetenschap, voor reproduceerbare proeven met uitkomsten die hun geldigheid behouden als de computer verandert."



Figuur 2: De Mona Lisa, opgebouwd uit 12 dominospelen (met elk 55 steentjes, van de dubbele nul tot de dubbele negen). De database 'lisa.dat' uit Knuth's Stanford GraphBase bevat een gedigitaliseerde versie van Da Vinci's beroemdste schilderij, met 360x250 pixels wier grijstint is vastgelegd in een cijfer tussen 0 (zwart) en 255 (wit). Ook in de Stanford GraphBase een programma dat uitgaande van zes of twaalf dominospelen, waarvan alle steentjes moeten worden gebruikt, een Mona Lisa opbouwt. Het minimaliseert de som van de kwadraten van de verschillen tussen de pixelwaarde uit 'lisa.dat' met de bijbehorende 0 of 255 van de dominosteentjes: 'assignment' in informatiecargon.

Vergaderziekte

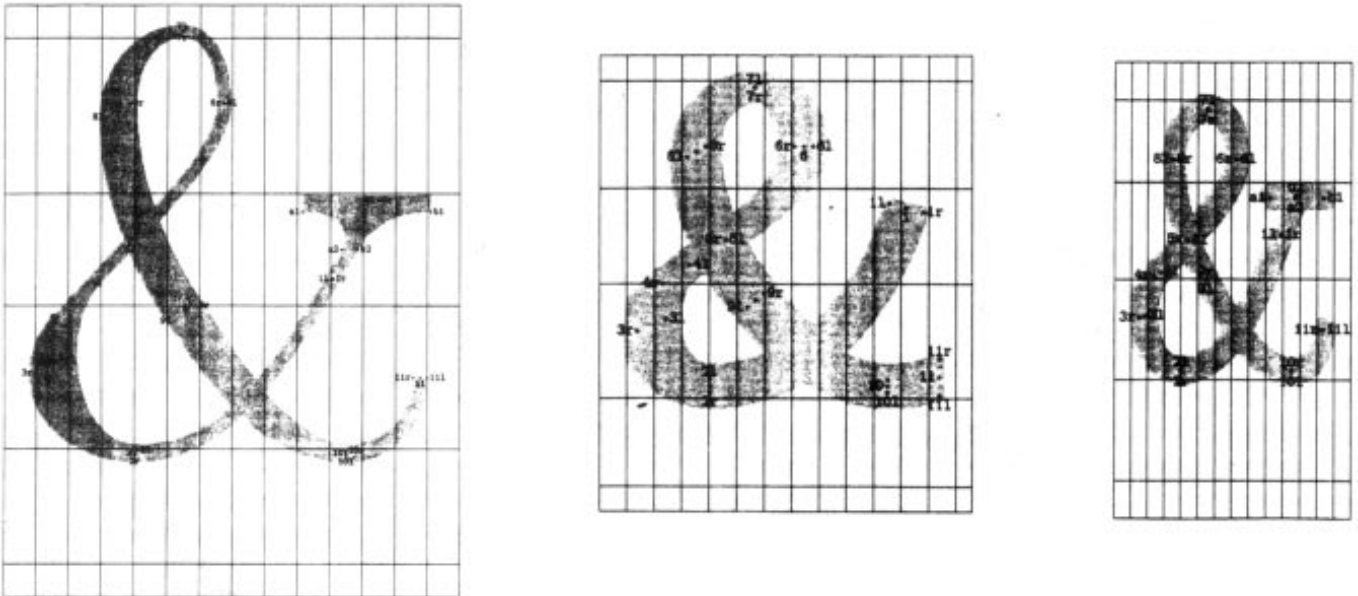
Vorig jaar ging Knuth op 56-jarige leeftijd met vervroegd emeritaat om zich, bevrijd van papierwerk en academische vergaderziekte, volledig aan de voltooiing van zijn levenswerk te kunnen wijden: *The art of computer programming*.

De oorsprong van dat project, een informaticabijbel waarvan tot nu toe drie delen zijn verschenen (met vertalingen in het Roemeens, Hongaars, Spaans, Chinees, Russisch en Japans), ligt in 1962. In dat jaar vroeg uitgever Addison-Wesley Donald, toen tweedejaars op Caltech (Pasadena), of hij zijn kennis van *compilers* (die hogere computertalen als *basic* of *fortran* vertalen naar 'machinetaal') niet te boek wilde stellen. "Toen ik die dag thuiskwam maakte ik een klasje met een indeling in twaalf hoofdstukken", herinnert Knuth zich. "Ik was net getrouwd en had geen idee van hetgeen ik me op de hals had gehaald. Programmeren lag me, tijdens een vakantiebaantje in mijn eerste jaar mocht ik 's nachts op een computer en was ik, moederziel alleen, direct verslaafd. Eerst had ik natuurkunde zullen doen, maar ik had een gloeiende hekel aan solderen. Informatica was nieuw."

The art of computer programming dijde spoedig uit. Behalve compilers wilde Knuth ook 'rangschikken' (*sorting*) behandelen, en algoritmes en optimalisering. In 1965 leverde hij een eerste manuscript in van 3.000 vel. "Ik had me er zwaar op verkeken. In overleg met de uitgever besloot ik tot zeven delen, met de compilers op het eind. Helaas is er na deel 3, verschenen in 1973, van alles tussengekomen. Intussen is het vakgebied sterk gegroeid en is revisie onvermijdelijk. Deel 4, over algoritmes, heb ik moeten splitsen in 4A, 4B en 4C. Als ik dat van tevoren geweten had, was ik er nooit aan begonnen. Nu kan ik niet meer terug, ook al omdat mijn ideeën door niemand anders fatsoenlijk zijn opgeschreven."

Veel meer dan door de Stanford GraphBase is *The art of computer programming* vertraagd door Knuth's activiteiten op het gebied van tekstverwerking en typografie. "Halverwege de jaren zeventig, toen het lood in de drukkerswereld plaats maakte voor offset, merkte ik dat de kwaliteit van de herdrukken van mijn boeken achteruit holde. Typografen hadden nauwelijks aandacht voor de speciale symbolen die in de exacte wetenschappen worden gebruikt en als drukkerszoon — er vloeit inkt door mijn aderen — leed ik daaronder. Ik besteedde de grootste zorg aan mijn boeken en wilde dat ze er mooi uitzagen. Toen ik in de gaten kreeg dat offsetmachines met digitale technieken werkten, wist ik dat er een taak voor mij was weggelegd."

Die taak resulteerde in het tekstverwerkingsprogramma \TeX (tau epsilon chi, naar de eerste letters van het Griekse woord *techne* dat techniek en kunstvaardigheid betekent) en het bijbehorende typografiesysteem METAFONT. Het geheel draait op praktisch iedere computer en is gedocumenteerd in vijf handleidingen en programmabeschrijvingen, bij elkaar zo'n 2.600 bladzijden. Bijna iedere wis- en natuurkundige schrijft zijn artikelen in \TeX . Knuth: "Ik begon in 1976 en dacht er met een jaar te zijn, het werden er tien. Ik houd niet van half werk. Ik geef om mooie curves, als scholier mocht ik graag met grafieken spelen. Die liefde keert in mijn typografie terug. Esthetische vormen in wiskunde vangen, daar gaat het me om. Ik herken mijn letters. Gisteren nog zag ik in de Leidsestraat een tram met Metafont."



Figuur 3: Voorbeeld van typografie volgens het computerprogramma Metafont. Ieder symbool is gekarakteriseerd door 62 parameters die aan of uit staan (zo dwingt monospace alle tekens dezelfde breedte op) dan wel bepaalde waardes aannemen (dikte van de stok, diepte van de komma).

Uitgangspunt voor Metafont was de ‘excellente’ typografie van (oudere) Addison-Wesley boeken, de *Proceedings* van de Koninklijke Nederlandse Akademie van Wetenschappen en het Zweedse tijdschrift *acta mathematica*. Ieder symbool, van gedachtenstreepje tot integratieteken, specificerde Knuth middels 62 variabelen. Iedere lettergrootte, ieder font (lettertype) kreeg zijn eigen instellingen. Eenzelfde onderliggende wiskundige structuur garandeert een visuele eenheid die bij handmatig ontwerpen onhaalbaar is. De gebruiker speelt met de knoppen, ziet het resultaat op scherm en beseft hoe moeilijk het is als amateurtypograaf een acceptabel alfabet te ontwerpen.

Ook T_EX zit vol geavanceerde wiskunde. Zo maakt het gebruik van een intelligent afbreekprogramma, dat niet aan één taal gebonden is, en het vult ingevoerde tekst niet per regel uit, maar per volledige alinea. Om dat te bereiken heeft Knuth een systeem bedacht dat afbrekingen of ‘overdreden’ spatiering vertaalt in ‘kosten’. Per alinea rekent T_EX van *alle* uitvoervarianten uit hoe duur ze zijn. De goedkoopste oplossing — een optimaliseringsprobleem waarvoor een Hongaars algoritme is geïmplementeerd — wint. Knuth: “Het kan zijn dat T_EX toch voor die lelijke afbreking kiest omdat *juist daardoor* nog lelijker wit op andere plaatsen wordt vermeden. Ik heb het systeem op alinea’s uit *Time* uitgeprobeerd en het verschil is tamelijk dramatisch.”

Een pagina per dag

Bevrijd van promovendi, congresbezoek en verplichte colleges richt Knuth al zijn energie op het vervolg van *The art of computer programming*. Dagelijks zit hij twee uur in de bibliotheek, tientallen artikelen tot zich nemend. Een e-mailadres heeft hij niet meer, wel een *homepage* op het world wide web, “een uitvinding invloedrijker dan de pil”. Ontspanning vindt hij achter het pijporgel (812 pijpen, 16 registers, 3 klavieren, gestemd volgens de regels van de Noord-Duitse barok) dat hij als zoon van een Lutheraans organist in zijn woning in Stanford heeft laten bouwen. Hij componeert en schreef in 1974 de novelle *Surreal numbers*, over een nieuwe manier om getallen te construeren, de enige keer dat een belangrijke wiskundige ontdekking eerst als fictie naar buiten kwam en pas daarna als artikel.

Daarvoor ontbreekt nu de tijd, *The art of computer programming* moet af. Knuth: “Een schrijftempo van een pagina per dag, dat is het streven. Concentratie is het sleutelwoord. Ik zal me moeten beheersen, alleen de tijdloze ideeën op schrift vastleggen. Informatica is zo opwindend dat ik het nauwelijks kan opbrengen naar bed te gaan. Twintig jaar heb ik nog te gaan, als het klaar is ben ik 77, laten we hopen dat ik het red. Kon ik de ontwikkelingen zolang maar stilzetten.”

$$\int_{-\infty}^{\infty} \left(\frac{\text{RAVEN}}{\text{Πάλλας}} + \heartsuit_{\tau\epsilon\lambda-\tau\alpha\lambda} \right) d\varphi \longrightarrow \left\{ \begin{array}{c} \text{THE} \\ \text{END} \end{array} \right\}$$

An Interview with Donald Knuth

DDJ chats with one of the world's leading computer scientists

Jack Woehr

Dr. Dobb's Journal*
editors@ddj.net

April 1996

For over 25 years, Donald E. Knuth has generally been considered one of the world's leading computer scientists. Although he's authored more than 150 publications, it is Knuth's three-volume *The Art of Computer Programming* which has become a staple on every programmer's bookshelf. In 1974, Knuth was the recipient of computer science's most prestigious prize, the Turing Award. He also received the National Medal of Science in 1979.

In addition to his work developing fundamental algorithms for computer programming, Knuth was a pioneer in computer typesetting with his \TeX , METAFONT, and WEB applications. He has written on topics as singular as ancient Babylonian algorithms and has penned a novel. Knuth currently is Professor Emeritus at Stanford University.

Born in Milwaukee, Knuth exhibited an early aptitude for patterns, as evidenced by his creation of crossword puzzles for the school newspaper. This ability culminated in the eighth grade when Knuth won a national contest sponsored by a candy manufacturer. According to Dennis Shasha and Cathy Lazere, authors of *Out of Their Minds: The Lives and Discoveries of 15 Great Computer Scientists*, the challenge was to compose as many words as possible using the letters in the phrase 'Ziegler's Giant Bar.' The judges had about 2500 words on their master list; Knuth came up with approximately 4500 words without using apostrophes.

Still, it was music that Knuth chose to study at Case Institute (later Case Western Reserve) until he was offered a physics scholarship. This lead him to his first encounter with a computer in 1956 — an IBM 650.

In this interview, Knuth chats with frequent *DDJ* contributor Jack Woehr about these and other topics.

— * —

DDJ: What distinguishes a 'computer scientist' from a 'computer programmer'?

DK: The difference between a computer programmer and a computer scientist is a job-title thing. Edsgar Dijkstra wants proudly to be called a 'computer programmer,' although he hasn't touched a computer now for some years. He wrote his really terrific essay on the *Humble Programmer* discussing this. To me, 'computer programmer' is an

honorable term, but to some people a computer programmer is somebody who just follows instructions without understanding what he's doing, one who just knows how to get through the idiosyncrasies of some language.

To me, a computer scientist is somebody who has a way of thinking, which resonates with computer programming. The way a computer scientist views knowledge in general is different from the way a mathematician views knowledge, which is different from the way a physicist views knowledge, which is different from the way a chemist, lawyer, or poet views knowledge.

There's about one person in every fifty who has this peculiar way of viewing knowledge. These people discovered each other at the time computers were born. There's a profile of different intellectual capabilities which makes somebody resonate, which makes somebody really in tune with computer programming.

There were computers in the 19th century, the 17th century... I imagine there are computer scientists in the pygmy forest. I haven't really carried this out as an experiment, but I imagine that people may not have machines but one in fifty of them, wherever you go, has this profile, this ability. I'm not a sociologist, nor an anthropologist, but reading publications, reading literature, I can sense how much people think like I do, even if they were writing from a different century.

This is the true explanation of why computer science became a university department so fast all around the world. The reason is not that computers are important tools for mankind, or something like that. The reason is that there were these people out there who had this way of thinking that never had a home before. They get together, they can communicate high bandwidth to each other, the same kind of analogies are meaningful to them. All of a sudden they could come together and work much more efficiently, not in someone else's territory that wasn't for them.

There was a time when there were no physics departments, there was 'Natural Philosophy,' which combined all kinds of different skills. Over the years, these strong areas of focus materialize, become recognized, and then they get a name. 'Computer Science' happens to be one of the more recent ones to crystallize in this way.

*This article was published in Dr. Dobb's Journal of April 1996, p. 16 – 22. Reprinted Courtesy of Dr. Dobb's Journal.

DDJ: When can we buy volume four of *The Art of Computer Programming*?

DK: I'm going to be putting it out 128 pages at a time, about twice a year over the next eight years. I'm estimating it now at a little more than 2000 pages. There will be volume 4-a, volume 4-b, and volume 4-c. Volume 4 in general is combinatorial algorithms. Volume 4-a is about finding all possibilities: There's a lot to be said about generating them in good ways — problems where finding all reasonable solutions is not a trivial task. 4-b is going to be about graph and network algorithms, and 4-c is about combinatorial optimization. So 4-a is 'find all arrangements,' 4-b is 'find arrangements that have to do with graphs and networks,' and 4-c is 'find the best arrangement.'

Into those 2000 pages, I have to compress about 200,000 pages of literature. I've been working on it a long time.

— — * — —

DDJ: Is the hiatus between volumes 3 and 4 writer's block, that you say, 'If I study this more...'

DK: No, that's a pretty good hypothesis. But I had started volume 4 and then realized I had to work on typography. There was a revolution in the printing industry. The printing industry became computer science. It changed from metallurgy to bits, to 0s and 1s. There was no way to print my books with the quality they had before.

I was going to take a year and give a computer scientist's answer to how to print books, and that took ten years. The systems are in common use today. So I think I'm going to be able to recoup that. It wasn't that I didn't have anything to say in volume 4, but that I had this other thing to do that was very timely. My students and I worked very hard on the desktop-publishing revolution.

— — * — —

DDJ: One goes to accomplish a task on the computer and realizes that to finish it requires another task, and to finish that one requires another...

DK: Niklaus Wirth always wanted to design airplanes, but he needed a better tool, so he designed many computer languages and microcomputers and so on. I needed to write my books in some way that would be immune to changes in technology, to capture the book in some electronic form that wasn't going to change when the printing establishment changed.

— — * — —

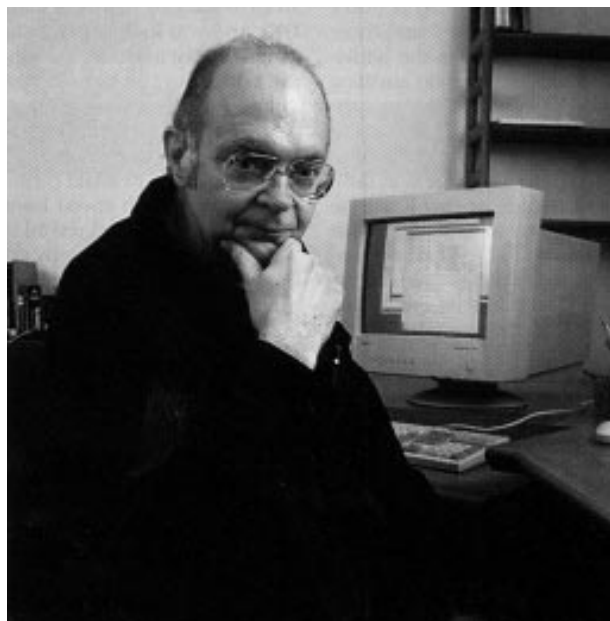
DDJ: I've read the laments of the memorizers of Egypt that were recorded by the scribes when introduced in the ancient kingdom. You're one who is not going to curse the darkness, you're getting ready for the time when books aren't printed anymore.

DK: I have books that are going to exist no matter how the technology changes.

— — * — —

DDJ: Has your digression into \TeX and METAFONT been profitable, as well as rewarding?

DK: I put it in the public domain, but I do get royalties on the books. I give one-third of those to the user group. The important thing, once you have enough to eat and a nice house, is what you can do for others, what you can contribute to the enterprise as a whole.



— — * — —

DDJ: I wonder if that same philosophy informs your scientific discipline. To the vast majority of the computer literate, you made a large contribution in stating once and for all how one, for instance, divides two binary numbers efficiently. A programmer wonders how to do something and reaches over his or her shoulder for 'knuth,' it's like reaching for a 'thermos.'

DK: I tried to write things up in a way that was jargon-free, so the nonspecialist would understand it. What I succeeded in doing is making it so that the specialist can understand it, but if I hadn't tried to write jargon-free, then I would have written for specialists, and then the specialists wouldn't be able to get it either. So I've been reasonably successful in boiling down a large volume of material, but still my books aren't easy reading except for specialists.

I'm about to publish a book, *Selected Papers in Computer Science*, which is a collection of papers I've written over the years for nonspecialists. It's being published jointly by the Center for the Study of Linguistics at Stanford (CLSI) and the Cambridge University Press. It has 15 chapters, each of which was an expository lecture. I enjoyed read-

ing them again, though I've edited them to take out sexist language! I think this book is something that might be of interest to the scientific community as a whole.

I plan eight books collecting all the papers I've written: There's going to be Selected Papers in Analysis of Algorithms, *Selected Papers in Digital Typography*, *Selected Papers in Fun and Games*, *Selected Papers in Programming Languages*, and so on. This is the second volume; the first volume was *Literate Programming*.

DDJ: What has the reward you offer for finding bugs in \TeX reached? I had heard it was up to \$40.96.

DK: Oh, that! The reward doubled every year until it reached \$655.36 and I froze it at 2^{16} pennies. It wouldn't take another ten or fifteen years before it began to exceed the GNP, you know! I paid out a couple of those last February.

— — * — —

DDJ: That was the problem posed by the inventor of the chessboard in ancient India, who asked for one grain of wheat on the first square, two on the second, four on the third...

DK: It was al-Biruni in 10th-century Baghdad who explained how to calculate 2^{64} efficiently by repeated squaring.

— — * — —

DDJ: This is one of the computer scientists of other eras about whom you spoke earlier.

DK: He was definitely a computer scientist. He knew how many grains of wheat there were without doubling 64 times.

al-Khwārizmī, who lived about 150 years before that, gave us his name as 'algorithm.' There were great books about chess already in the 9th century. The successor to Haroun al-Rashid of the *Thousand and One Nights*, Caliph al-Ma'mun, established a great center of learning and a 25-year flowering of art and science — al-Khwārizmī published there about algebra and arithmetic and geography. One of his books is about the Jewish calendar. I discuss this in *Selected Papers in Computer Science*.

— — * — —

DDJ: I understand you are not entirely a partisan of the C++ language.

DK: C++ has a lot of good features, but it has a lot of dirty corners. If you don't mind those, and you stick to stuff that can be counted well-portable, it's just fine. There are many constructions that are ambiguous, there's no way to parse them and decide what they mean, that you can't trust the compiler to do. For example, you use the 'less-than' and 'greater-than' signs not only to mean less-than and greater-

than but also in templates. There are lots of little things like this, and many things in the implementation, that you can't be sure the compiler will do anything reasonable with.

Languages keep evolving, and that's necessary. I find it impossible to write books for archival without resorting to the English language, though. Whatever computer language is in fashion, you can guarantee that within a decade or two it will be completely out of fashion. In my books, I try to write things that aren't trendy, but are things that are going to be worth remembering for other generations. I'm trying to distil what, in my best judgment, out of thousands and thousands of things that are coming out now, is most deserving to be remembered.

— — * — —

DDJ: You've mentioned Edsger Dijkstra. What do you think of his work?

DK: His great strength is that he is uncompromising. It would make him physically ill to think of programming in C++.

— — * — —

DDJ: There's a great difference between his scrupulous examination of each and every algorithm, and the practice of commercial programming, where megabytes of code with known and unknown bugs are thrust at the user.

DK: I know, I know...I'm somewhere in the middle. I try to clean up all bugs and I try to write in such a way that, aware of the formal methods, I can be much more sure than I would be without the formal methods. But I don't see any hope of proving everything in these large systems, because the proof would be even more likely to have bugs!

— — * — —

DDJ: Programs nowadays depend on huge bodies of code that aren't written by the main author.

DK: And you look at them and see how each spends most of its time trying to defeat the other. It's all these black boxes you can't open, so you have to build your own firewall.

This is nothing new. In the '60s, someone invented the concept of a 'jump trace.' This was a way of altering the machine language of a program so it would change the next branch or jump instruction to retain control, so you could execute the program at fairly high speed instead of interpreting each instruction one at a time and record in a file just where a program diverged from sequentiality. By processing this file you could figure out where the program was spending most of its time.

So the first day we had this software running, we applied it to our Fortran compiler supplied by, I suppose it was in those days, Control Data Corporation. We found out it was spending 87 percent of its time reading comments! The

reason was that it was translating from one code system into another into another.

— — * — —

DDJ: GUIs haven't made this any better.

DK: We now have so much more processing power that the only people who see what's happening are people writing game programs, who need real high-speed animation. I got new software at Christmas, and I'm really discouraged by the number of failures that I noticed. I'm giving up the idea of using this software to get much done. I'm going to go back and write my books with good old reliable Emacs and \TeX .

I still haven't found an editor on the Macintosh where I can create a 1-byte file that has the letter 'a' in it that I can send to the PC and read on an Emacs-like editor. I got optical-character-recognition software that has a choice of 50 output formats. Each one of them included all kinds of font-changing mechanisms, and so on. Finally, I found one called 'database text' and if I output it in database text, that was what I was expected to get — raw ASCII characters.

It's a natural way to get job security, to make computer systems that need one's expertise.

My \TeX system is trying to go the other way, so I wouldn't have to go through the professional typesetters, the professional font designers. I could still use these professionals, but I could use them to provide added value. I didn't have to go through a level of writing something for them to do and then check on it. I can write something, and they can tell me how to improve, but I don't need to write something that they already have on the menu.

— — * — —

DDJ: Is the profile of a programmer (which we were discussing earlier) one of an individual who needs this sort of control?

DK: That's an interesting concept, the need for power! I've always thought of it more in other terms, that the psychological profiling is mostly the ability to shift levels of abstraction, from low level to high level. To see something in the small and to see something in the large.

When you're writing a program, you're saying, 'Add one to the counter,' but you know why you're adding one to the counter. You can step back and see a picture of the way a process is moving. Computer scientists see things simultaneously at the low level and the high level.

The other main characteristic of the computer scientist is dealing with nonuniformity of discreet, nonuniform things. We have ten cases instead of one. We don't have one law of thermodynamics, or something. We have case one, case two, case three — each is different. We build models out of nonuniform parts. We're so good at that, we don't see sometimes that a uniform part would work, if it would. But

people who are only good at the uniform things could never build a model out of nonuniform parts, could never do the things that a computer scientist does, because they have to find a single rule that's going to cover everything.

We have this aesthetic of cost, how much work it takes to do things. If your mental model is APL, you optimize in different ways than if your mental model is a RISC machine.

— — * — —

DDJ: When you look back at the first three volumes of *The Art of Computer Programming* is there anything you would change?

DK: I have about 900K of changes to the first three volumes, plus changes to other books that I've written, that I plan to make available on my Web page. There are four kinds of changes, and the different kinds are distinguished typographically.

One kind is a 'bug' and that means that I have to correct something that is technically wrong. One kind is an 'amendment,' which means that there is some goodie that deserves to be in there. One kind is an 'improvement,' something which would go in a future edition of the book, but is probably not worth people's writing it in their own copy. The fourth thing is called a 'plan,' something still under construction, where the picture is changing so fast that I don't think it's cost efficient for me to write on, since I'll just have to do it again, the kettle is still boiling, but I wish to state that I intend to retool something in a certain way.

It will probably be a while before publishing changes so that entire books can be available online. I don't know how to convert the present system to a better one that will still have the proper incentive structure. There's something all fouled up about the way that software is compensated and font designers are compensated and musicians are compensated, and other intellectual-property rights issues. A scientist can be supported by the National Science Foundation but a font designer is not supported by a National Font Foundation. Both of them are doing things that contribute to the public good.

— — * — —

DDJ: Is this just an expression of a love for symmetry, or is there a social injustice being performed here?

DK: I think that the fact that somebody's expertise is in the field of beauty and graceful strokes and another's is in the field of integrals and differential equations shouldn't mean they have completely different ways of getting paid.

The Free Software Foundation people are putting out good stuff. It's hard for the untrained person to wade through the jargon to install it. Richard Stallman and I don't agree all the way down the line, but he can be an effective arguer! Stallman is one of my heroes, of course. He probably likes some of the things I do, too!

It offends me when people get patents on trivial stuff that we would expect any student to do. I come from a culture where the compensation came because one's work was recognized as advancing civilization. Of course, in literature there were royalties, not grants. But mostly it was that people had done good work, so you figured they deserved a continuing job. If I were to consider a strategy of becoming rich, it would be so I could support people who need support, who I consider are doing things for the future, like the dukes and duchesses used to support Mozart.

— * —

DDJ: If you could climb in the pulpit and scold, exhort, and encourage every working programmer in the United States, what would you tell them?

DK: The first thing I would say is that when you write a program, think of it primarily as a work of literature. You're trying to write something that human beings are going to read. Don't think of it primarily as something a computer is going to follow. The more effective you are at making your program readable, the more effective it's going to be: You'll understand it today, you'll understand it next week, and your successors who are going to maintain and modify it will understand it.

Secondly, ideas that are mathematical in nature should be the property of the world and not of the individual who thinks of the theorem. I'd prefer that all but the most sophisticated algorithms be made public and that everybody use them, and not that every time you use such-and-such a method you should pay a nano-penny to some fund.

I wrote an open letter to the head of the U.S. Patent Commission, published in the current printing of the CWEB manual. I said, 'What if lawyers were to have rights to their

precedents? What if people had patents on words of the English language, and every author who wanted to write a novel would have to check which words they were using and pay royalties to the owners of those words? Can't you see how obvious it is that the quality of the legal system and the quality of published books would go down? Because you're taking away the building blocks that people need to do their job.'

The basic building blocks that software designers need to do their jobs are algorithms and languages and mathematics. It's traditionally impossible to patent a mathematical formula, for very good reason. Anyone who would wish to calculate the area of a circle and use πr^2 should have to pay a royalty for that: It's exact, it's a universal thing. I think that algorithms should be in exactly the same category. Algorithms are mathematics.

Algorithms are the building blocks to create large, useful systems. The service that you're providing for people is making those systems more accessible, packaging them better, giving better help on the phone, but not just having a method that other people could put into another system.

I would encourage programmers to make their work known the way mathematicians and scientists have done for centuries. It's a comfortable, well-understood system and, you get a lot of satisfaction knowing people like what you did. The whole thing that makes a mathematician's life worthwhile is that he gets the grudging admiration of three or four colleagues.

Acknowledgments

The author wishes to acknowledge the help of Steven R. Wheeler of Vesta Technology in Wheat Ridge, Colorado, in preparing for this conversation with Dr. Knuth.

Knuth meets NTG members

March 13th, 1996

Abstract

On January 6th 1996, Kees van der Laan informed the NTG that Donald Knuth would be in Holland in March. Knuth was invited by the *Mathematisch Centrum* (MC, nowadays called *Centrum voor Wiskunde en Informatica*, CWI) because of CWI's 50th anniversary. Both Knuth and Mandelbrot were invited as speakers at the celebration.

The NTG noticed that this was an exceptional occasion to organize a special meeting with Knuth for all Dutch T_EX and METAFONT users who would like to meet the Grand Wizard himself.

Fortunately Knuth accepted the NTG invitation and so a meeting was organized in 'De Rode Hoed' in Amsterdam on March 13th. About 35 people from all over the country and even from Belgium joined to meet Knuth.

Everything was recorded on both video and audio tape by Gerard van Nes. Christina Thiele volunteered to write this transcript.

Erik Frambach: Welcome, everyone. This is a very special meeting on the occasion that Mr. Donald Knuth is in Holland. The NTG thought it would be a good idea to take the opportunity and ask him if he would be willing to answer our questions about T_EX, METAFONT, and anything else connected to the things we do with T_EX. Luckily, he has agreed. So we are very happy to welcome Mr. Donald Knuth here — thank you for coming.

Tonight we have time to ask him any questions that we have long been waiting to pose to him [laughter]. I'm sure that all of you have many, many questions that you would like the Grand Wizard's opinion about. So, we could start now with questions.



Donald E. Knuth: I get to ask questions too! [laughter] Last Saturday I was in Prague and the Czechoslovak T_EX users had a session something like this and you'll be glad to know that I saw quite a few copies of 4T_EX CD-ROMs at that meeting.

It's not my first time in Amsterdam. I was in Amsterdam in 1961, so it's only been 35 years, and probably less than 35 years till the next time. I guess they're tape recording these

questions-and-answers to try to keep me honest, because they also did that in Prague. So in case the same question comes up, you'll have to take the average of the two answers. [laughter]

Wietse Dol: Did you know that Barbara Beeton does that? She mails you and says "Tape everything."

Knuth: Yes, that's what they said in Prague too! [laughter] I think she's desperate for things to do, or maybe she just has a lot of questions. But before I open questions, let me say that one of the most interesting questions asked me in Prague was after the session. And I wish it would get into [the record]. The question was: how did I meet Duane Bibby, who did the illustrations for *The T_EXbook* and *The METAFONTbook*? I always wanted people to know about that somehow.

Here's the story. I had the idea that after writing math books for many years, I wanted to have a book that had more weird — well, anyway, different — illustrations in it. Here I was writing a book about books — books have illustrations, so why shouldn't I have illustrations too. So, I wrote to an artist called Edward Gorey. Does anyone know ...

Frans Goddijn: Yup. *Amphigorey*. Beautiful.

Knuth: Yes, Edward Gorey. *Amphigorey*. He makes very morbid drawings but with a wonderful sense of humor. I had used several of his books with my children. I thought he would be a natural person. I wrote him two letters but he never responded. Then I wrote to a Japanese artist called Anno, Matsumasa Anno, who is really the logical successor to Escher. [...] Ano does what Escher did but in color, so I asked him if he would do it. He sent me back a nice letter, saying "I'm sorry I don't have time because I have so many other commitments, but here are five of my

books full of pictures and if you want to use any of those, go ahead.” I wanted personalized pictures.

Then I went to a party at Stanford where there was a lady who worked for a publisher. She’d just met a brilliant young artist who she’d just worked with. I invited him to come to my house, and we spent some time together and he’s a wonderful person. Duane lives now up in northern California, about 4 hours’ drive from my house, so I only went up there once to see him. He sometimes comes down to the San Francisco area on business. First we discussed the book and then he sent me a bunch of drawings and all kinds of sketches that he had. Originally, \TeX was going to be a Roman, and he drew this man in a toga with olive branches in his head — which is why the lion has the olive branches now. But all of a sudden he started doing sketches of his cat, which really seemed to click, and pretty soon he had a draft of all 35 or whatever drawings, using a lion. Most of those eventually become the drawings in the book, and we adjusted half a dozen of the others. When I went to visit up at his house, I got to meet \TeX the cat. He looks very much like the one you see in the book. So that’s the story about Duane Bibby.

Erik: Thank you. Who would like to start with the first question? Please identify yourself when you ask one.

Piet van Oostrum: My name is Piet van Oostrum. You have this wonderful lion on the *\TeX book* and the lionness on the *The METAFONTbook*. What about baby lions?

Knuth: Oh, I see ... [laughter]. Duane still does illustrations for special occasions. He’s made illustrations for the Japanese translation of both the *\TeX book* and the *The METAFONTbook*. He has \TeX and META both dressed up in Japanese costumes. So now, if there happens to be some kind of an offspring that would come out of some other user, then, I imagine he would glad to help do it. But it would probably be a little bit of an illegitimate child, from my point of view [laughter]. I mean, I wouldn’t take responsibility for anything those characters do [laughter].



Piet: So what are your ideas about the offspring of \TeX and METAFONT?

Knuth: Well, I think that no matter what system you have, there will be a way to improve it. If somebody wants to

take the time to do a good, careful job with it, then as we learn more about typesetting, it will happen that something else will come along. I personally hope that I won’t have to take time to learn a new system, because I have enough for my own needs. But I certainly never intended that my system would be the only tool that anybody would ever need for typesetting. I tried to make it as general as I could with a reasonably small program, and with what we knew and understood about typesetting at the time. So these other projects — I don’t consider that they’re a threat to me or anything. I hope that there will be some compatibility so that — I mean, I’d like to be immortal — so that the books I’ve written now could still be typeset 50 years from now without having to go through the files and edit stuff. I like the archival and machine-independent aspects of \TeX especially, and I tried to set a model, a minimum standard of excellence for other people to follow.

Hans Hagen: but when you look in the future, ... you consider today’s programming by a lot of people as an art, well a lot of art takes hundred of years to be recognized as art. In about a hundred years there will be pretty different computers, the programming languages will be changed, the media on which we put all those things will be changed. Real programs and everything related to them, will they ever have a chance to become immortal, as you see it?

Knuth: Did you state your name? [laughter]

Hans Hagen: I’m Hans Hagen.

Knuth: You’re saying that it’s pretty arrogant of us to assume that what we do now will last at all. Technology is changing so fast that we have absolutely no idea what people are going to think of next. One hundred years ago, physicists were saying there was nothing more to do in physics, except to get another decimal — a fifth decimal place for the fundamental constants — and then that would wrap up physics. So, there is no way to know about these things. But I do believe that once we have things in electronic form and we have mirror sites of them, there is a fair degree of immortality — whereas paper burns.

Do you know anything about this project called ‘The Clock’, being developed by Stewart Brand and his colleagues? He’s the one who published the *Whole Earth Catalogue*. They have a bunch of people that are considering if they could build something that would last for a thousand years ... I don’t want to go on too much more about that. I do hope that the stability of \TeX will make it possible to reproduce the things we’re doing now, later. And since it’s fairly easy to do that, I think it will happen — unless there’s a nuclear holocaust. Some mathematicians have this debate about the Platonic view ... does everything in mathematics exist and we’re just discovering it, or are we actually creating mathematics? In some sense, once something gets put into bits, it’s mathematics and therefore it exists forever, even if the human race dies out — it’s there, but so what?

Erik: Who’s next?

Marc van Leeuwen: If I could extend a bit on the previous questions. The stability of \TeX itself, I could imagine, might be a stumbling block for development of new things exactly *because* it's so stable and everybody's already using it. So if something comes along that is just a bit better, then people will not tend to use that because it's not available everywhere, and there are all kinds of reasons to keep on using the old thing.

Knuth: I guess I said in Florida that people are still trying to use the old fonts that I'm still trying to stamp out from the world. Four years ago I redesigned the Greek lowercase delta and I made the arrowheads darker. I didn't change anything in the way \TeX operates — all the dimensions and the characters' heights and widths stay exactly the same. But I did tune up a lot of the characters. Still I see lots of math journals are still using the old ones from four years ago, and I get letters and preprints from people with the old-style delta. I changed it because I just couldn't stand the old versions [laughter]. Now I've got home pages — if I ever have some errata to \TeX or something I put them there: <http://www-cs-faculty.stanford.edu/~knuth>. This gets to my home page, and there's a reference saying, 'Important notice for all users of \TeX ', and that page says 'Look at the lowercase delta and if you have the wrong one, you die!' [laughter]

I understand that people have a reluctance to change from something that they've become accustomed to. I know of two main successors to \TeX : one is $\varepsilon\text{-}\TeX$ and the other is NTS. $\varepsilon\text{-}\TeX$ is going to be apparently 100% compatible with \TeX , so if somebody doesn't switch over to incompatible features, then they have a system that still works with old things. That will allow a gradual change-over. It'll take more space on a computer, of course, but that's not a big deal these days. The people who work on $\varepsilon\text{-}\TeX$ always sent me very reliable comments about \TeX when they caught errors in my stuff, so I imagine they're going to be doing a careful job. So it'll be one of these things where you walk into a random installation of UNIX or whatever and you'll find $\varepsilon\text{-}\TeX$ there as the default, and you'll still have \TeX . Then you also have certain other features that might be really important to you for your special applications.

Johannes Braams: You mentioned $\varepsilon\text{-}\TeX$ and NTS. But are you also aware of the Omega project?

Knuth: Oh, the Omega project? Yes, I'm hoping to use that myself for the authors' names in my *The Art of Computer Programming*. I've been collecting the names of Chinese, Japanese, Indian, Hebrew, Greek, Russian, Arabic authors and I want to typeset their names properly [laughter], not just in transliteration. I have some rudimentary software that will do this for proofing purposes, for getting my database going and for writing to people and saying, 'Is this your name?' With the Omega system, I'm hoping that it'll be accompanied by good fonts that will make it possible for me to do this without a whole pile of work. Right

now, to get the Arabic names, I have to use Arab \TeX , to get the Hebrew names ... I had a terrible time trying to get Hebrew fonts on CTAN two weeks ago — I can tell you that whole story if you want to know ... I kept clicking on the different things and they would refer to files that didn't exist and README files that were four years out of date and inconsistent, so I couldn't find any Hebrew fonts. Maybe you have it on your CD ...

Johannes: I could certainly point you to someone who could help you with the Hebrew font — I know someone in Israel who's trying to do Hebrew support within the Babel system. And they do do typesetting in Israel with \TeX .

Knuth: My own typesetting friend in Israel is Dan Berry, who unfortunately is fairly committed to troff [laughter]. I'm sure that I can get good Hebrew through Yannis and Omega. I sure hope UNICODE is going to arrive sooner rather than later; it's much better than the alternatives for much the reasons that Marc [van Leeuwen] mentioned. I haven't found a great enthusiasm in Japan for UNICODE, because they have a system that seems to work pretty well for them, so why change. Everytime I ask a Japanese for his name in UNICODE, he'll say, 'what's UNICODE? Here's my JIS name'. But the JIS characters don't include the Chinese codes, and in fact, my own name — I have a Chinese name — and my name in JIS isn't quite the same. There are two different UNICODE characters, one for the Japanese version and one for the Chinese.

In the back? Kees?

Kees van der Laan: I have a lot of questions of course. But I would like to start with some questions about METAFONT. The first one is: how come macro writing in \TeX and METAFONT is so different?

Knuth: Why are macros in \TeX and METAFONT so different? I didn't dare make \TeX as extreme as METAFONT. These languages are of completely different design. METAFONT is in some ways an incredible programming language — it's object-oriented macros. You have macros in the middle of record structures.

The way I designed these languages is fairly simple to describe. Let's take \TeX . I wrote down one night what I thought would be a good source file for *The Art of Computer Programming*. I took a look at Vol. 2, which I had to typeset. I started out on the first page, and when I got to any copy that looked very much like something I had already done I skipped that. Finally I had examples of all the different kinds of typesetting conventions that occur in Vol. 2. It totalled 5 printed pages — and you can even see these pages — exactly what my original test program was — in a paper by David Fuchs and myself, where we talked about optimum font caching.¹ In there, we gave an example and we show these 5 pages, which would illustrate what I wanted \TeX to be able to do. I wrote out what I thought I would like to type — how my electronic file should look. And then, I said, OK, that's my input, and here's my output

¹ACM Transactions on Programming Languages and Systems 7 (1985), 74.

— how do I get from input to output? And for this, well, it looks like I need macros [laughter].

Same thing for METAFONT. I went through my first draft of all the fonts that later became Computer Modern. I wrote actually in SAIL, an Algol-like compiler language, but SAIL had a macro ability, so I developed a few primitive macros in which I could say, ‘pick up the pen’, ‘draw from point 1 to point 2’, and things like that. These macros were compiled by the SAIL compiler into machine language, which would then draw the letters. I went through the entire alphabet, and by the end of the year, I had some 300 little programs, each one drawing a letter. Then I realized what kind of a language I would want to write in, to describe the letters. So one day, on a family camping trip — I was in the Grand Canyon with my wife and kids — I took an hour off, sat under a tree and wrote out the program for the letter A, in a language that I thought would be a good algebraic language, reflecting at a high level what I had been doing with pretty primitive low-level instructions in my SAIL programs. I did the letter B, too. Capital A and B, and then went back to the camping trip. These sheets of paper where I have my original programs are now in Stanford’s archive — the program for the letter B was published in a Stanford library publication called *Imprint* last year. The woman who’s in charge of rare books and manuscript collections at Stanford is quite interested in METAFONT so she wrote a little article about what they have.

That program again implied that I wanted some macros to go with it. But these needed to be much more structured than the macros of \TeX . It had to be that when I said, ‘z 1 prime’, this would actually be equivalent to ‘(x 1 prime, y 1 prime)’ and I wanted to be able to write, ‘z 1 prime’ without any delimiters. It turned out that in order to have a high-level language that would feel natural to me writing the program, it had to look completely different from \TeX . So \TeX and METAFONT share a common format for error messages and certain other data structures inside, but otherwise, they’re quite different systems because, in order to have a good high-level language, I don’t want to have to waste time writing parentheses, brackets, commas, and other delimiters.

Kees: It’s a nice introduction to my second question [laughter]: For the future of MetaPost, which allows mark-up of pictures, with `.eps` as the result, what is your attitude to 2.5d for MetaPost and METAFONT? For example, adding a triple as an analogy of the paired data structure?

Knuth: MetaPost already has a data structure for triples because of color. So RGB are actually triples of numbers.

Kees: Yes, but the triple as a data point in space?

Knuth: Ah, I see. I did write METAFONT in a way that has hooks in it so that it can be easily extended; [for example], if you want to draw 3-dimensional pictures, for perspective and projective geometry instead of affine geometry. The program itself for METAFONT was written so that it could easily be changed by people who wanted to have a

system that goes beyond the basics. I always wanted the systems that I would make widely available would be able to handle 99% of all applications that I knew. But I always felt there were going to be special applications where the easiest thing would be to change the program, and not write a macro.

I tried to make the programs so that they would have logical structure and it would be easy to throw in new features. This hasn’t happened anywhere near as often as I thought because people were more interested, I think, in inter-changeability of what they do; once you have your own program, then other people don’t have it. Still, if I were a large publisher, and I were to get special projects — some encyclopaedia, some new edition of the Bible, things like that — I would certainly think that the right thing to do would be to hire a good programmer and make a special computer system just for this project. At least, that was my idea about the way people would do it. It seems that hasn’t happened very much, although in Brno I met a student who is well along on producing Acrobat format directly in \TeX , by changing the code. And the Omega system that you mentioned, that’s 150,000 lines of change files [laughter]. I built in hooks so that every time \TeX outputs a page, it could come to a *whatsit* node and a *whatsit* node could be something that was completely different in each version of \TeX . So, when the program sees a *whatsit* node, it calls a special routine saying, ‘how do I typeset this *whatsit* node?’ It’ll look at the sub-type and the sub-type might be another sub-type put in as a demo or it might be a brand-new sub-type.

Similar hooks are in the METAFONT program. If people have extra time when they’re not browsing the Web [laughter], I recommend as a great recreation to read the program for METAFONT. Some parts of it are pretty rough going and I hope that nobody ever finds a bug there because I’d hate to have to look at them [laughter]. But those are the rasterization routines, the things that actually fill in the pixels. There are many other things in that program — the linear equation solver that it has and the data structure abilities ... lots of beautiful algorithms are in there — to take square roots in fixed point, and the intersection of two curves, and so on. METAFONT is full of little programs that were great fun to write and that I think are useful and interesting in their own right. I think when John Hobby wrote MetaPost, he enjoyed it, because he could add his own nice little programs to the ones that are already there.

I’m a big fan of MetaPost for technical illustrations. I don’t know anything that’s near as good, so I’m doing all the illustrations of *The Art of Computer Programming* in MetaPost. Also, the technical papers I’ve written are going to be published in a series of eight volumes by Cambridge University Press, and all the illustrations, except the photographs, are going to be MetaPosted. The first volume of these eight was the book, *Literate Programming*; the second volume is going to come out this summer and is going to be called *Selected Papers in Computer Science*. It reprints a dozen or 15 papers that I wrote for general audi-

ences, not for specialists in computer science, but in *Scientific American* or *Science* magazine and things like that. The third volume will be about digital typography, and it'll reprint all my articles in *TUGboat* and things about \TeX . What do you think, by the way — should I publish in that third volume the memo that I wrote to myself the first night, when I designed \TeX ? I put it in a computer file and it's in the archives, but I've never shown it to anyone. [round of "of course!" and "sure" and laughter from the audience]. Maybe it'd sell more books [more laughter].

Frans Goddijn: You need to put it on your home page and we can then decide —

Knuth: No, no. That way we'd never sell the books [laughter]. Not that I'm a mercenary type of person, of course. It's in a file called \TeX — well 'teks', actually. I have to admit I pronounced it 'teks' for a month or two — I was thinking of 'technical texts', though. `tex.one` was the name of the file and it would make interesting reading probably, someday.

And your name is?

Jan Kardan: In this company I will probably ask a very heretic question, but a little heresy makes a lot of fun — talking about METAFONT. There are probably many type foundries now [that] crank out lots of good-quality fonts and kerning tables. It's not clear whether PostScript or True Type will survive. Do you think that METAFONT will survive text fonts? Not talking about the math fonts.

Knuth: I don't think the extra capabilities of METAFONT have proved to be necessary for good-quality type fonts, although I think that you can still make better-quality type fonts with it. Designers find it difficult to think as a computer person does, in the sense that when people in the computer business automate something, trying to make the computer do something, it's natural for us to have parameters and say that we're going to try to solve more than one problem. We try to solve a whole variety of problems based on the parameters that people set. But it's much easier if people gave us only a single problem with a single parameter, then we could have the computers do exactly the prescribed thing. Computer scientists have become accustomed to thinking of how we would change behavior as conditions change, but designers aren't at all accustomed to this. They are much happier if the boss says one month, "Give me a roman font," and the next month, "Give me a bold font." It's much more difficult to say, "Show me how you would draw something no matter how heavy I want the letters to be." METAFONT provides a way to solve that problem and to draw characters with parameters, but it's a rare designer who's comfortable with that notion. They can do multiple master fonts by making multiple drawings and then matching up points between the drawings and hav-

ing the computer interpolate. The multiple master fonts in PostScript allow up to four parameters, and almost all of them have only one or two parameters. The most I know of is two; probably others have gone all the way to four. But then they have to provide drawings for all the extreme points of these parameters.

In spite of this limited use of parameters, what's available commercially is quite beautiful, as far as readability is concerned, although it doesn't really provide the quality that you guys had in the Netherlands in the 17th century. What's the man's name, the great punch cutter at Enschedé — he made 4.5, 5pt up to 16pt, and each letter was designed for its size, and fonts had a nice uniform appearance. This wouldn't have happened at all with the Type1 fonts. There were two guys who did most of the punch cutting for Enschedé and others in the 18th century: One of them, Fleischman, was a genius for really beautiful letters; the other, Rosart, was just good at making lots and lots of letters [laughter].² [...] They were fun. Rosart would make all kinds of highly decorated alphabets and things like that. I have a big coffee-table book that gives examples of all the fonts from Enschedé, which was translated into English by Matthew Carter's father. Anyways, in this book, *Typefoundries in the Netherlands*, you can look at these typefaces and weep.³

Still, on a laser printer, we get pretty good fonts now, and therefore it looks like there won't be that many professional type designers using METAFONT. Pandora was a good design by a genuine graphic artist. METAFONT has turned out to be wonderful for making ordered designs and special-purpose things for geometry. There's now this really neat system in Poland where they have \TeX and METAFONT in a closed loop — \TeX outputs something and then METAFONT draws a character and if that doesn't fit, \TeX says, 'go back and try it again'. Jackowski and Ryćko understand \TeX and METAFONT, and the programs are well documented and can do these things. So Metafont isn't going to disappear for that reason; but it's never going to be taught in high school.



²Johann Michael Fleischman, 1701–1768; Jacques-François Rosart, 1714–1777.

³*Typefoundries in the Netherlands from the Fifteenth to the Nineteenth Centuries*, by Charles Enschedé, translated by Harry Carter (Haarlem: Stichting Museum Enschedé, 1978), 477 pp. This magnificent book was composed by hand and printed by letterpress to commemorate the 275th anniversary of Joh. Enschedé en Zonen.

Frans Goddijn: My name is Frans Goddijn and I have one question with some sub-questions [laughter]; I'd like to ask the sub-questions first. What I'm wondering — and this may have been asked often before — is whether you would consider, in retrospect, what you have created [to be] an art or a tool? And the reason I ask is — when I hear you speak with so much passion for type fonts and the beautiful algorithms that you put into METAFONT that you would like to point people to and the recognition that you get from people who understand that — but, there is a vast majority of users who just got T_EX from some server, never realized who created it, and use it to typeset not always very pretty documents [laughter]. They do that in a very crude way and don't care less. You froze T_EX at a certain point, allowing other people to build around it. I was wondering how such a thing would feel to a father — are you father of a piece of art that other people use as a tool, or is it a child that you have frozen in its development, that will never grow up ... there are so many questions ... if you just go back to the art vs. tool idea, and your feelings about that.

Knuth: Obviously, if I write something that has a lot of power to do many different things, it'll be possible to make it do awful things. I just came from the *Rijksmuseum*, where they have an exhibit called "The Age of Ugliness". It was a whole bunch of fancy silver bowls from the late 19th century When you say an art, I'm not sure I understand exactly what you mean. To me, art is used in two quite different senses, most often nowadays in the sense of fine art, while art, originally, *Kunst*, was anything that was not natural — so we have the word *artificial*, something that is made by people instead of by nature. The Greek word is *techné* [laughter]. But then you refer to a tool as something that is maybe just a device that is the fastest way to get from here to there but maybe you don't care about elegance ... But what I think you mean when you talk about art is the aesthetics — something about beauty and something with a little bit of love in it. With T_EX, my idea was to make it possible to produce works that you are proud of; I assumed that people can enjoy actually spending a little extra time making the results better. I didn't expect that the whole world would be doing this [laughter].

Incidentally, I can't understand the mentality of a person who writes graffiti on a beautiful building although I can see why drawing is fun. Why would you want to scrawl something — some kind of animal instinct of territory might account for it, I suppose, but it's really impossible for me to conceive of such actions.

When it comes to matters of aesthetics, you can't dictate taste. You can't say that your idea of beauty is going to match anyone else's idea of beauty. But I did want to have a tool where we could reach the highest levels of beauty according to our own tastes. I didn't allow people to have letterspacing very easily, but I tried to make everything else easy [laughter]. Of course, I originally designed T_EX just for myself, for *The Art of Computer Programming*; I thought my secretary and I were going to be the only users. And it wasn't until later that I was convinced that I should

make it more general and so on. But I did want a tool for myself by which I could produce books that would make me feel good after spending almost all my life writing those books.

I started writing *The Art of Computer Programming* when I was 24 years old and I still have 20 years of work to do on it. That's a lot of time. I don't want to write those books if they're going to come out looking awful. I wanted a way to make it possible [to produce good-looking books]. Originally, when computers started out, they knew only numbers, digits. The 19th-century computers could print tables. Then we had computers that could do numbers and letters, but only on a teletype machine; so you had some capital letters and a 32-character set. But then, after I graduated from college, we got ... let me see, I was probably ten years out of college before we could do lowercase letters on a computer. You know, the PASCAL language, when it came out, it used all uppercase letters — there was never any consideration that there would be more than 64 characters in a computer's repertoire. Finally, we were beginning to see in the middle 70s that computers could actually do lowercase letters, and produce something that looked a little bit readable, a little bit like books. Wow! [laughter].

Then there was this development of typographic software starting at MIT in 1960 and going through 4 or 5 generations, leading to troff and EQN, where there was even mathematics being typeset. In 1977 I therefore knew an existence theorem: It was possible to typeset something that looked almost like good mathematics. EQN was being used in physics journals and experience showed that secretaries could learn how to do it. So I thought, "Why not go all the way to the end, to convergence?" What I wanted to do with T_EX was not to be a little refinement over troff and the other things, but I was saying now, "Let me try to go to the best typography that's ever been achieved by mankind" Except for the illuminated gold leaf type of lettering, I wanted to at least — when it came to black and white printing — I wanted to match the best conventions that had been achieved. Computer typesetting had gone through this lengthy development, getting a little better and a little better. It was time to say, "Well, let's jump to the end now." Of course, I didn't think this would be an activity that everybody would want to do. But there were enough people that would care about trying to get as much quality as possible, that they could be — well, that's why I finally made T_EX more available. The American Math Society were the first people, nearly the first people who convinced me that I should make the system do more than I originally intended.

Andries Lenstra: Why didn't you start from troff? It was completely inappropriate?

Knuth: Yes, yes. You see, troff was patched on top of ... I mean, there was a whole system, it was a fifth generation, each of which was a patch on another one. So it was time to scrap it and start all over again: "Here's what the language should be, so let's design some good data structures

for it.” Not “Let’s try to be compatible.” I had the advantage that I was not at Bell Labs, where I wouldn’t be hurting anybody’s feelings by saying, “Let’s throw it all away” [laughter]. It was impossible for the people at Bell Labs to do such a thing — it wouldn’t be nice. But it occurred to me that now that we had proof that this goal was possible, I should start over, and rethink how I could get from input to output, so the program could be much more unified, much smaller, and would also work. I mean, troff was collapsing all the time. A lot of the earliest users of T_EX had been frustrated by troff breaking over and over again, so it had gotten unwieldy. But it had also proved that there was light at the end of the tunnel.

I also had to scrap T_EX, you know, and start over again; after five years, I decided that it would be best to go back and re-do the program. But it would have been very hard to do that if my friend in the next office had done it [laughter]. So, I just have this philosophy that there will be always some people who are more interested in quality than others, and I wanted to make T_EX good for them. I don’t see any good way to make it impossible to make a bad document, unless you have only a system with a small menu of options; that’s good for a large class of users, to make a system that’s so simple that you can’t possibly do anything ugly in it.

Erik: I think it’s time for a coffee break now — we’ll take five or ten minutes.

Knuth: Johannes, you had a question that you had to ask, so let’s get that over with [laughter].

Johannes Braams: It’s about typesetting. What is your opinion about the skyline model of typesetting? In T_EX, you talk about boxes: each letter is inside a box, and we glue boxes together to a line, and the line itself is inside a box, and each line is viewed as a box and the boxes are fitted together to form a paragraph. The skyline model tries to go a little bit further than the rigid box and line, and tries to take into account that some of the descenders in the upper line and the high parts in the lower line don’t overlap, so that you could actually have lines much tighter together — especially in math typesetting, that could be an advantage.

Knuth: Hmm, I guess you’re talking about general principles of computer graphics where you have rectangles inside a picture, instead of having the rectangles grouped only inside of a rectangle. ... This certainly would be a major change in all the data structures of T_EX. You could go to a quad tree structure or something like that. All the things that people use to solve hidden-line problems and do rendering, to find out what’s in front of something else, and all the algorithms they use to make movies like *Toy Story*. It would be most valuable, I imagine, for catching unusual cases in math formulas.

I have two feelings about these things. One is that I like to see people extending the things that computers can do automatically. People learn a lot when they try to do this. The

whole field of artificial intelligence has been one of the areas that has had greatest spin-offs to computer science because they’ve tried to solve very hard problems. Especially in the early days, they came up with methods that turned out to be useful in many other parts of computer science. So, it’s my feeling that when people are working on more ambitious goals, they develop powerful techniques that often have very relevant spin-offs. Even so, after they’ve solved that problem, they’re going to think of something else which will be another refinement and so on — they’ll never have a situation where they’re going to automatically create the most beautiful document. There’s going to be a time when you can look at the output and see that you can still improve it. Designers of the most automatic systems would be well advised to at least still leave a chance for somebody to move something up and down and fake out their automatic algorithm.

The philosophy that I had when I did T_EX was that I would try to have a system that did 99% of everything automatically, and then I would look at what remained and I would kludge the rest. But kludging it is one way to say it; another way of saying it is “Tidy up the rest,” or “Dot the i’s and cross the t’s.” My feeling is that this non-automatic part gives me a little extra pride that I have put the spit and polish on the final product, that I know I did it. If it occurs a lot, then it’s a nuisance and I’m wasting time. But if I can really limit this to 1% — if I’ve spent 30 hours writing a paper and it takes me only another 15 minutes to clean up, then I’m happy to do another 15 minutes at the end. It’s a small little extra that gives me a chance to celebrate the fact that I’ve finished the paper.

The spacing that T_EX does worst right now, in my experience, is with respect to square root signs being a little too tight, with the operand either too close to the radical sign or too close to the bar line or both; I find that I’m most often fiddling with that. I’ve adopted in the book *Concrete Math* and also in *The Art of Computer Programming* now, the convention where in the math formula I put an @-sign where I want one math unit of extra space. The @-sign is then defined to have a math code of hexadecimal 8000, which means that this will invoke in math mode and the @-sign will be regarded as a macro that adds one math unit of space. So I’ll type ‘square root of’ ‘@-sign’ ‘log of n’ [laughter], because otherwise the space before ‘el’ is a little bit too tight. Now maybe even this skyline model wouldn’t know that ‘el’ was too tight, maybe it would. But it’s cases like that ...

The most common case really is where I have something like ‘x squared over 3’, where you have a simple superscript and then a slash, and then the denominator. There’s almost always too much space before the slash. And this is true, I find, in all the books that I used to think were typeset perfectly by hand [laughter], but now I’m sensitive to this. Now I go through, typically with *emacs*, and look for all occurrences of something with a one-character exponent followed by a slash, and most of those look better with

a negative thinspace before the slash. It would be nicer if I didn't have to do that. But still, it's a small thing for me.

Would the skyline model help me much? Sometimes I run into cases where I'll add another word to the answer to an exercise in order to avoid a clash between lines. Here, the lines are actually not getting spread apart too far, but they're so close together that the 'subscript k less-than-or-equal to n' will clash with a left parenthesis in the next line. And I don't want the type to be quite so close together there. Now, if I had been smarter, I would have designed my \leq -sign to have a diagonal stroke under the $<$ instead of a horizontal bar, and I wouldn't have had those clashes — too late for that now. [laughter]

Kees?

Kees: May I ask you a question about your attitude to mark-up in general? And let me illustrate it by first telling a story. When we started with using \TeX etc., we mean actually we start with \LaTeX — I mean, that is the effect in Holland. And then I looked at the products of the mark-up and I did not like it. And then I was wondering, what is your attitude to that? I'm sorry to say so, I paged through the \TeX book file `texbook.tex` and I looked at all the things in there and then I thought, "Well, I have some idea of what your ideas are of mark-up." And when you explained about METAFONT and all those things not in there, which you have implicit — am I wrong if I summarize this, that you adhere to something like minimal mark-up?

Knuth: Yes. For example, when I am reading Edsger Dijkstra's books, every time I get to a section where it says 'End of Comment', it strikes me as redundant. And I always think, "Oh, yes, this is Edsger's style." When I wrote a paper for his 60th birthday, I said at the end, "Acknowledgment, I want to thank Edsger for such-and-such," and 'End of Acknowledgment' [laughter].⁴ But that's the only time in my life I'll ever do that. Maybe I'm an illogical person, but apparently half the people using `html` now type only the `p` at the beginning of a paragraph, and the other half type only a `/p` at the end of a paragraph [laughter]. Hardly anybody uses both, according to what my spies tell me. And I don't know what the heck these systems actually do with the unbracketed material. When I write `html`, I'm scrupulous with my mark-up. If you look at my home pages — I'll pay you \$2.56 if you find any case where I started something and didn't close it with the right tag. I tried to be very careful in that, and to indent everything very well, and so on. But I found it a terrible nuisance, because it's not the way I think.

I think a high-level language, to me, is something that should reflect its structure in some visual way but not necessarily explicitly; so that, when I know the conventions, we can suppress some things. Parentheses are one such convention and mathematics got a lot better when people invented other notations like operator precedence that we can see structure without spelling it out in too much detail. A mathematician spends a lot of time choosing notations

for things, and one of the things we try to avoid in mathematics is double subscripts. I read one French PhD thesis where the author had five levels of subscripts [laughter]— he kept painting himself into a trap. He started out with a set x_1 through x_n , so then when he talked of a subset, it had to be x_{sub-i_1} through x_{sub-i_k} , and then he wanted to talk of a subset of this, so then he had a theorem that says, let a sub-b sub-c ... and so on [laughter]. I try to choose notations that give me the economy of thought at a high level.

That's why I probably didn't believe in a great deal of mark-up in the \TeX book; I would begin typewriter type and end typewriter type for sections by saying `\begintt` and `\endtt`. I would also delimit the lines and when I'm presenting parts of the plain \TeX macros, `\beginlines` and `\endlines` — those macros are in the file, since it's very important to me to see how that works. But in other cases, I've left [things] as simple as possible, for me to see visually the beginning and end of stuff. It's something also like problem solving — sometimes, if I've solved a problem and I'm not worried about it anymore, I forget to tell anybody else the solution. I was always a very bad committee chairman because I'm not very good at finishing that last ending line, I guess. Still, with `html`, the document was short and I decided that my home pages were going to be used by many different kinds of browsing software so I had better be very rigorous.

While I was developing \TeX , I attended one of the meetings of the committee that designed SGML and had a very good discussion with Charlie Goldfarb and the other people on the committee — we only had that one meeting near Stanford. Certainly I appreciate the fact that this structure makes it possible to build other kinds of programs around what you have. The more structure you have in a document, the easier it is to make a database that includes things about it, and knows what's going on. I never objected to it; I just always felt that in order to maximize my efficiency, I didn't want to mess around with full mark-up unless I had to.

X: SGML allows minimizations; that's why the end-paragraph is not necessary. So that's one of the reasons why it's so difficult sometimes. You have a formalization to minimize.

Knuth: But \LaTeX doesn't allow it.

Johannes: But we do have some books, however, permitting omitted end-tags in \LaTeX 3, but that's not far enough along.

Knuth: Well, talk to him [laughter]. I don't need a special editor for `html` — people are hyping fancy things where you can click on a tool and it'll put in the start and end tag together — but when I wrote my files, I did make up a simple `emacs` macro that would take whatever tag I just typed and create the end-tag. All it had to do was search back till it found a less-than sign and then copy that string twice and

⁴*Beauty is Our Business* (Springer, 1990), 242.

put a slash in front of it, so I used that all the time — it was easy.

Johannes: Quite different type of question now, from someone who'd like to ask here: literally, he writes, "Why is the height of the minus sign in the cm symbol font the same as the height of the cmr plus sign?"

Knuth: Ah. A lot of people are wondering about that one. Where you have 'a minus c' or you say 'x sub minus' or something, why is it that the height and depth are greater than the actual shape of the minus sign?. In fact, it's not just the plus and minus, it's also the +, -, ±, ∓, ⊕, ⊖, ⊗, ⊙, × and ÷ — if you look at the code for these, there is a `\beginarithchar` macro that begins all of the arithmetic characters in the font, guaranteeing that they will have the same size.

Johannes: But it doesn't say why.

Knuth: That's right — it doesn't say why. And the reason is that early on, I wanted certain things to line up the same. For example, if you had

$$\sqrt{x+y} + \sqrt{x-y},$$

I wanted the square root signs to be place in the same way. Otherwise, you would get

$$\sqrt{x+y} + \sqrt{x-y}.$$

[slightly exaggerated here to show the point] And so there are many other cases where you have formulas where there's a plus sign in one part of a formula and a minus sign in the other part, and for consistency of spacing, it ought to look symmetrical. There are other cases, I readily admit, where you have only a minus sign — you never have a similar thing with a plus sign, and you wonder why there's extra space left there. So I say `\smash minus` [laughter] in those cases.

Johannes: The particular application, why this question was asked — Michael Downes from the AMS —

Knuth: Yes, Michael Downes, he has more experience than any of us in this room; he's the chief typesetter of most of the mathematics in the world.

Johannes: He has a problem properly attaching a superscript on top of the `\rightarrowfill`...

Knuth: The `\rightarrowfill`? OK ... The `\rightarrowfill` is this thing that makes a right arrow of any desired length, and then he wants to put a superscript on this. What's the macro for building that up? I haven't used that page in a long ... [laughter]⁵ The `\rightarrowfill` is made up of minus signs and so probably if I had known Michael ... known about that in

the old days, I would have changed the plain `\TeX` macros so that it would not use the height of the minus sign in the `\rightarrowfill` operator [...] ⁶ Anyways, I've now told you the reason why it's there for the other ones.

Johannes: Another question, which is about multiple languages. There's a problem when you have one paragraph where you have different languages.

Knuth: Yes, the `\lccode` changes. This is the ...

Johannes: And I've been told that inside one paragraph you can only use one hyphenation table, which is the one which is active at the end of the paragraph. So, switching hyphenation tables inside paragraphs. Suppose, for example, you have a paragraph with English text, with a German quote inside it, the German quote being several lines long.

Knuth: I know that `\TeX` will properly keep track of which hyphenation table to use. The glitch, the mistake, that I didn't anticipate is if the two languages have different `\lccode` mappings — so that each has a different idea of which characters are lowercase. When you hyphenate, you need to hyphenate an uppercase word the same as an lowercase word, so `\TeX` uses the `\lccode` of a character to convert every letter into the lowercase code of that letter. I didn't anticipate that people might, for different languages, have a different mapping from uppercase to lowercase. And so it's that mapping that, at the end of a paragraph, applies to all the languages in the paragraph. But otherwise, `\TeX` is careful to keep track of what language you have.

And by the way, there's a file called `tex82.bug`. Go to the CTAN archives, and find subdirectory `/knuth`, and under that `/errata`, and that's where this is. At the end of `tex82.bug` this particular error about `\lccode` is mentioned as being something that's an oversight that's too late to fix.

Marc van Leeuwen: Why is it too late to fix? It would conflict with other things?

Knuth: Yes. So that people are already using these things in lots of documents, and it's very hard to change. In fact, I don't see any way to fix it [laughter]. I would say that when you are faced with a situation where you're doing multiple languages with multiple `\lccodes`, this is a good reason to write your own version of `\TeX`.

Andries Lenstra: Could I ask a question? Happily enough, I'm not the first person to mention `\LaTeX`, so I may mention it now. There's a situation that often arises when people try to write a PhD thesis where they want to change `\LaTeX` code because they think they know better about things of beauty or typography, and unhappily enough they are not experts on `\LaTeX`, so they don't succeed or they succeed badly. In general, people who know

⁵Knuth was trying to remember `\buildrel`; see *The TeXbook*, p. 437.

⁶In fact, the `\leftarrowfill` and `\rightarrowfill` macros now omit the height and depth of the minus, in `plain.tex` version 3.14159 (March 1995).

about typography can't write beautiful L^AT_EX code or other forms of code, and vice versa — people who know how to write these forms of code, are no experts on typography. What do you think of the endeavors in the past to bring the two worlds together, for instance, as Victor Eijkhout has tried to do with his `lollipop` format, a machine to create other formats. I would have thought that it would have had a big success but the opposite seems to be the truth. What do you think of it?

Knuth: I'm not familiar with the details of `lollipop`. I suppose that was based on a famous quotation from Alan Perlis, who said that, "If somebody tells you that he wants a programming language which will only do the right thing, give him a lollipop."

Andries: Yes.

Knuth: I'm sure that the lollipop effort was instructive and worthwhile, but I don't know the details so I can't answer in great detail on this. Probably the type designers didn't find the language easy to learn. I do think that we're having much more communication now, as every month goes by, between the people that know about type and the people that know about macros. It's just a matter of time as we wait for these waves to continue moving — we're nowhere near a convergent stage, where T_EX has reached its natural boundary and the type designers have reached their natural boundary. They're still moving toward each other. I don't think it's like a hyperbolic geometry, where they never will get together.

The main difficulty of course is that T_EX is free, and so a lot of people will say, "Well, how could it be any good, if you're not charging money for it?" A lot of the people in the type design community would only work in things where there's money behind it; money proves to them that it's worth talking to people. So it just takes a little while till they see some good examples, which will make them more open for these discussions. And that's happening all the time in different countries.

In the Czech Republic I was quite delighted to learn that the new encyclopaedia in Czech, which is the first one for many years, is being done with T_EX. And not only that, it's being done with a very high budget. They made this decision because they tried all the other systems and were disgusted with them. They had good results with T_EX. Many other commercial publishers are using it too because they talk to their friends at the big publishing houses. This will, I think, be solved with time. And products like `lollipop` are very worthwhile in the meanwhile to facilitate this. It takes time to bring different communities together. I think the financial factor is definitive for a lot of people.

Piet van Oostrum: I don't know if you have ever looked into the L^AT_EX code inside, but if you look into that, you get the impression that T_EX is not the most appropriate programming language to design such a large system. Did you ever think of T_EX being used to program such large systems

and if not, would you think of giving it a better programming language?

Knuth: In some sense I put in many of the programming features kicking and screaming, and I'll try to explain the background. I know how Leslie went about writing L^AT_EX — first he would write the algorithms out in a high-level programming language, with **while**'s and **if-then**'s and so on, and then he would pretty much mechanically convert this to T_EX macros. If I had suspected that such a style was going to be the most common use of T_EX, I probably would have worried a lot in those days. Now, computers are so fast that I don't worry so much about the running time, because it still seems to go zip!

In the 70s, I had a negative reaction to systems that tried to be all things to all people. Every system I looked at had its own universal Turing machine built into it somehow, and everybody's was a little different from everybody else's. So I thought, "Well, I'm not going to design a programming language; I wanted to have just a typesetting language." Little by little, I needed more features and so the programming constructs grew. Guy Steele began lobbying for more capabilities early on, and I put many such things into the second version of T_EX, T_EX82, because of his urging. That made it possible to calculate prime numbers as well as do complicated things with page layout and figure placements. But the reason I didn't introduce programming features at first was because, as a programmer, I was tired of having to learn ten different almost-the-same programming languages for every system I looked at; I was going to try to avoid that. Later, I realised that it was sort of inevitable, but I tried to keep it as close to the paradigm of T_EX as a character-by-character macro language as I could. As I said before, I was expecting that the really special applications would be done by changing things in the machine language code. But people didn't do that, they wanted to put low-level things in at a higher level.

Piet: What do you think, for example, of something like building in a programming language which is, from a software engineering point of view, easier to use?

Knuth: It would be nice if there were a well-understood standard for an interpretive programming language inside of an arbitrary application. Take regular expressions — I define UNIX as "30 definitions of regular expressions living under one roof." [laughter] Every part of UNIX has a slightly different regular expression. Now, if there were a universal simple interpretive language that was common to other systems, naturally I would have latched onto that right away.

Piet: The Free Software Foundation is trying to do that and Sun is trying to do it and Microsoft is trying to ...

Knuth: The Free Software Foundation is trying actually to include also the solutions of Sun and Microsoft. In other words, to make all of the conventions work simultaneously as much as possible. And that conflicts with my own style, where I've tried to have unity rather than diversity ... I did-

n't go for ten ways to do one thing. C++ is similar — when the committee would say, “Well, we could do it this way or this way,” they did both. I hadn't gone that route in my system, because it is messy. But I admit that the messy way is the best that can presently be realized in practice.

Marc van Leeuwen: I have a question about literate programming. I know you must be very fond of it, if I understand your interviews —

Knuth: Yes, I'm so fond of it that I could ... well ... OK [laughter]. You know, I'm really so fond of literate programming, it's one of the greatest joys of my life, just doing it.

Marc: My question was that obviously it's not nearly as popular as TeX is, and, what's more, there isn't much coherence in the world of literate programming. There are a dozen different systems being used — some people favor this, some people favor that — and this worries me a bit. I too am very fond of this style of programming, but I would like to see it being used much more.

Knuth: Literate programming is so much better than any other style of programming it's hard to imagine why the world doesn't convert to it. I think that Jon Bentley put his finger on the reason and it was something like this: There aren't that many people in the world who are good programmers and there aren't that many people in the world who are good writers, and here we are expecting them to be both. That overstates the case but it touches the key point. I think that everyone who's looked at literate programming agrees that it's a really good way to go, but they aren't convinced that ordinary students can do it. Some experiments at Texas are proving otherwise, and I've had a smaller-scale experience at Stanford. It's a hypertext way of programming and I imagine that with better hypertext systems that we're seeing now and people becoming so familiar with the Web, we're going to get a variety of new incompatible systems that will support literate programming. Hopefully somebody with time and talent, and taste, will put together a system of literate programming that is so charming it will captivate a lot of people. I believe that the potential is there, and it's just waiting for the right person to make that happen.

Marc: I think one problem might be that if you compare your programs with the average program that people write, there just aren't nearly as many interesting algorithms in the average program, so literate programming doesn't add too much to a program which is very dull by itself.

Knuth: Well, thank you for your comment. But maybe sometimes I make a non-interesting algorithm interesting just by putting in a joke here or there. I've taken programs that I got from Sun Microsystems, for example, and as an exercise, spent the afternoon converting them to a literate form. There weren't any exciting algorithms in there, but

still, you could look at the final program and it was better — it had better error diagnostics, better organisation, it corrected a few bugs. I don't have time to go over to Sun and show them this, and say, “Why don't you rewrite your operating system?” [laughter] but I know that it would be much better. So all I ever published was the very simple rewrite of the WC word count routine in UNIX, which is not at all an exciting algorithm, but as a demo of how it could be done.⁷

My approach to literate programming isn't the only one, of course, and in the recent book by a group at Princeton, *A Retargetable C Compiler*, Chris Fraser and Dave Hanson used a variety of literate programming to describe their C compiler. Other books are coming out now that are using some flavors of literate programming. I was talking to someone at Microsoft who said that he thought literate programming was on the rise, and I said, “Does that mean the next version of Windows is going to be all done in literate programming?” “Well, no, not exactly” ... [laughter]. The people who've experienced literate programming will never go back, and they'll probably gain influence gradually. The companies that use it are going to sell more products than their competitors, so pretty soon this will happen. I imagine that there are about ten thousand users of literate programming and a million users of TeX, so it's a factor of a hundred.

Marc: Do you think it still has to develop? I get the impression that with so many tools around, that it's not yet mature. The idea is mature, but the implementation still has to ...

Knuth: Yeah, that's true. There's great need for programming environments based on this idea. It's not at all easy to create these environments and to have the power to promote them and maybe the support to do it in a way that wouldn't make it too expensive or too hard for people to install. The most ideal thing would be if the Free Software Foundation were to adopt it, or something like that, or some of the people they work with. Actually, [Richard] Stallman [of the FSF] designed a variant of literate programming that he uses, and he has it well integrated with TeX, in his own style. He hasn't put it in too many of his programs, but he has a version. It's one of these things that needs, as you say, to mature.

Marc: Do you believe it should go in the direction of integrated systems, where you really have all the facilities you need in one system? Because I think the tendency is more towards very minimalistic systems that do not do any pretty printing because that gets you into too much trouble when you're switching programming languages. So it really boils down to something which is very flexible but not very convenient for someone to use.

Knuth: One programming language is good enough for me so I'm not the right person to ask. But then, I guess, for the *The Art of Computer Programming*, for the next twenty

⁷D.E. Knuth, *Literate Programming* (Stanford, 1991), 341–348, based on a prototype by Klaus Guntermann and Joachim Schrod, *TUGboat* 7 (1986), 135–137.

years, I'm pretty sure that CWEB is going to be as good as anything I need. I'll write programs for Mathematica and I'll write some programs for MetaPost; I could develop or use literate programming for those programs, but I don't think I will. I don't write so many lines that I would gain a great deal ... although I would get a better program afterwards. Unless somebody already presents me with a good system for it, I won't go ahead with MathWeb or MPWeb. But with CWEB, I'm going to write an average of five programs a week for the foreseeable future, and there, my productivity is infinitely faster when I do it with literate programming.



One other thought flashed in my mind as I was talking just now ... I wrote a paper last year, I think it was, about mini-indexes for literate programs⁸ and here I was trying to show what sort of programming environment would help me. In the listings for \TeX the program and \METAFONT , and also for the Stanford Graphbase, on the righthand page of each two-page spread, there's an index to all the identifiers used on that page and where they were declared. My paper explains the system I used to get those indexes, and this kind of functionality would also be needed in any hypertext system. These minimalistic systems are attractive primarily because a good programmer can write them in a couple of days, understand them and use them, and get a lot of mileage out of them. Once somebody writes a good hypertext system for literate programming, I think that'll attract a

lot of people — a system that doesn't crash, and has a familiar user interface because it's like other hypertext systems that we're already using. The time for that will be ripe in about two years.

Erik: It's half past nine now and I think we'll have to stop here. I want to thank our special guest, Donald Knuth, for the time here. I think we've all learned a lot now. We're very happy that you were able to be here. Thank you very much.

Knuth: I really appreciate all the work you did to get this special room here on rather short notice. [applause]

Erik: Also, thank you to Elsevier Science, who helped, in the person of Simon Pepping; and your English colleague, Sebastian Rahtz, who is not here, although I expected him. But he paid for the coffee and tea, so thanks. There's of course a little present that we have for you. I hope you like it! [He presents a book about Dutch art called 'De Stijl'.]

Knuth: Yes, ... the Dutch type designer, Gerard Unger, came to Stanford for three weeks, he and his wife Marjan, and they talked about things like this to our type designers. They also related fashion of clothes and furniture and architecture to type styles as well. This is great. This was done with \TeX ?

Erik: I don't think so ... as we are in Holland now ... [laughter]. [He also presents a pair of wooden tulips]

Knuth: A nice gift for my wife.

Erik: And of course a copy of the Euro \TeX '95 proceedings. [He presents the proceedings]

Knuth: Oh!! I thought you'd never — [laughter]. Yes, I was looking at this last week in the Czech Republic, so thank you everyone.

Erik: What is your opinion about the fonts we used?

Knuth: I think it's ... oh, you used the Computer Modern Brights. Yes, the only complaint I had was that the kerning in the word ' \TeX ' itself could be tuned a little bit.⁹ It's quite attractive — thank you very much.

⁸Software Concepts and Tools 15 (1994), 2–11.

⁹“The \TeX logo in various fonts,” TUGboat 7 (1986), 101.

T_EX without 4T_EX on the PC

Siep Kroonenberg
Economics Department
Rijksuniversiteit Groningen

22 april 1996

A bare-bones setup

In spite of my occasional involvement in the 4T_EX project and the existence of a 4T_EX installation at our department, I am not a 4T_EX user. Ever since I started using T_EX at the end of 1991, I have used my own configuration files in combination with the existing 4T_EX installation: I wanted to be able to run the compile–preview–print cycle, preferably from the commandline, with only a minimum of re-configuration of my system.

If that is also what you want, you may want to look at the batchfiles in the `\rawemtex` subdirectory of the 4T_EX workbench cd. However, these only cover emtex itself.

Currently I use a single batchfile `texcfg.bat` of my own which does all the configuring. It defines some environment variables and DOSKEY macros, making use of the pre-existing configuration files. After it has run, the core components of T_EX can be run with a simple command–and–filename commandline.

It does nothing particularly clever and doesn't offer the functionality of 4T_EX, but it does – contrary to the `rawemtex` batchfiles – also integrate PostScript support and several utilities. Adopting it may help you in the task of fitting all the pieces together.

The CD version is available at www.eco.rug.nl/medewerk/siep/texcfgcd.zip. The new edition of the 4allT_EX CD will contain a matching version. Several colleagues in my department use a similar batchfile adapted to the T_EX-installation on our LAN.

How far do you get?

Supported are the compiler, emtex and PostScript dvi-drivers with support for Metafont automatic font generation, BibT_EX, MakeIndex and Ghostscript. Essential options are taken care of, but additional options will be passed on to the respective programs.

Block-compile is not explicitly supported but requires only a little bit of manual setup: create a small auxiliary T_EX document `frame.tex` consisting of

1. the preamble of the document you are working on, up to and including the line `begin{document}`
2. a line `\input tmp.tex`

3. a line `\end{document}`

When you want to compile just a fragment of your document, save that fragment as `tmp.tex`. You can write a macro for this. Then compile `frame.tex`. If you work under Windows you might want to edit and compile in separate windows so that your document can remain loaded in your editor during compilation.

The most important 4T_EX features which are not supported and are non-trivial to set up are automatic font generation for Type 1 fonts and graphics conversion facilities. You *can* use Type 1 fonts and graphics with the `texcfg` configuration, if you use the PostScript dvi-driver `dvips` and don't mind previewing and maybe also printing with Ghostscript. `Dvips` can use Type 1 fonts directly and can include graphics in eps- or Encapsulated Postscript format. Since virtually any graphics program can export directly in eps-format, and eps is even for bitmaps often the preferred format, this is no limitation.

Ghostscript runs quite well on current PC hardware, but you do sacrifice some convenience with previewing and some quality with printing if you don't have a PostScript printer.

Installation and customization

Copy `texcfg.ba_` to a file `texcfg.bat` on your hard-disk, editing it as necessary. See the comments in the file itself for details. As noted there, you also need a sufficiently large environment and for `doskey` a sufficiently large buffer. Finally, as is also indicated there, you need to place a copy of Ghostscript's dos extender `dos4gw.exe` on your path.

I have tried to make basic customization easy: there are six clearly marked places where changes may be necessary to get it working, and many more where customization is optional. If you have a PCL- or PostScript printer there probably is already a DOSKEY macro which you can activate by uncommenting.

Additional customization may be desirable if you want to use fonts or style files of your own, and also if you don't keep all sources for one document in one directory, as is the assumption of `texcfg`. `Texcfg` contains ample comments to guide you in such customization.

It is fairly trivial to replace the doskey macros with 4dos aliases or with regular batchfiles. You might want the latter for fully automated T_EX runs since doskey macros can't be called from batchfiles.

1 Ghostscript under Windows

If you want to use Ghostscript's Windows executable, gswin or gswin32, you can add an icon for it in the usual way with Windows' Program Manager. You can find it in the same directory as the DOS executable, i.e. in \emtex\utils\ghostscr. It doesn't require any additional configuration beyond what is already done.

In the same directory you will find gsview[32], which adds a proper Windows interface to gswin. When you start it for

the first time it will tell you how to configure it. Be forewarned though that gswin runs much better by itself.

2 Dvips

I couldn't entirely avoid custom configuration files since the dvips Metafont mode can only be set in a configuration file. Texcfg creates at startup a printer-specific or rather resolution-specific configuration file %printer%.cfg in the root directory of the c:-drive with the proper Metafont mode.

Since this configuration file had to be created anyhow, I also added a line p +MYPSPMAP.MAP to facilitate the use of additional Type 1 fonts. You can safely ignore this feature if it means nothing to you.

```
@echo off
:: TeX from the command-line version for 4allTeX 3rd edition CD1:
:: 4TeX workbench 20 april 1996

:: Create or erase TeX configuration:
:: EMTEXDIR undefined or 0: create
:: EMTEXDIR defined and <>0: erase

:: !!!! Check AT LEAST the settings for !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
:: - EMTEXDIR (drive letter cd-rom)
:: - initial loading of doskey with enlarged buffer
:: - LATEX (209: old or 2E: new)
:: - PRINTER (printer resolution for POSTSCRIPT printing;
::   laserjet for 300dpi, ljIV for 600dpi)
::   irrelevant for PCL printing
:: - doskey tprint (PCL printer choice: print to file or to printer)
:: - doskey gprint (printer choice)
:: !!!! These items are marked like this !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
:: This batchfile might place a file ljiv.cfg in c:\
:: Limitations
:: -----
:: - 86- en -286 versions not supported
:: - Automatic font generation supported only for MetaFont fonts
:: - memory management: is o.k. if either TeX is run from a Windows DOS
::   box, or in CONFIG.SYS [Q]EMM386 is called with a RAM parameter
::   (for expanded memory). In the first case RSX is used, in the second EMX.

:: !!!! BOOT FILES !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
:: CONFIG.SYS:
:: The SHELL command should specify a sufficiently large environment, eg.
::   SHELL=C:\COMMAND.COM C:\ /E:1000 /P
:: AUTOEXEC.BAT:
:: If doskey is already loaded there, it should specify a sufficiently
:: large buffer, eg.
::   lh doskey /bufsize=1000
:: Also, copy EMTEX\UTILS\GHOSTSCR\DOS4GW.EXE to a directory on your path.

:: BEGINNING OF CODE ::::::::::::::::::::::::::::::::::::::::::::::::::::
SET OPT=
IF %EMTEXDIR%[==[ SET OPT=INSTALL
IF %EMTEXDIR%[==0[ SET OPT=REINSTALL
IF %OPT%[==[ GOTO ERASE

:INSTALL
:: !!!! drive letter cd-rom !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
SET EMTEXDIR=d:\EMTEX
SET EMTEXDIRSLASH=d:\EMTEX\

IF %OPT%==REINSTALL GOTO PATHDONE

:: !!!! Uncomment if doskey not loaded previously !!!!!!!!!!!!!!!!!!!!!!!
:: lh doskey /bufsize=1000

:: Addition to search path
PATH %PATH%;%EMTEXDIRSLASH%METAFONT

:PATHDONE
SET OPT=

:: ENVIRONMENT VARIABLES ::::::::::::::::::::::::::::::::::::::::::::
:: memory manager: RSX requires Windows DOS box; EMX requires EMM
SET RSX=%EMTEXDIRSLASH%RSX.EXE
SET EMX=%EMTEXDIRSLASH%EMX.EXE

:: !!!! old or new LaTeX !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
:: SET LATEX=209
SET LATEX=2E

:: TeX input files (.STY); may add own directories
IF %LATEX%!=2E! SET TEXINPUT=%EMTEXDIRSLASH%LATEX!;%EMTEXDIRSLASH%INPUTS!;%EMTEXDIRSLASH%LATEX209!
IF %LATEX%!=209! SET TEXINPUT=%EMTEXDIRSLASH%LATEX209!;%EMTEXDIRSLASH%INPUTS!;%EMTEXDIRSLASH%LATEX!

:: bibtex input files
```

```

SET BIBINPUT=.

:: precompiled TeX formats
SET BTEXFMT=%EMTEXDIRSLASH%BTEXFMTS

:: TeX font metrics; may add own directories
SET TEXTFM=.%EMTEXDIRSLASH%TFM!
:: same for dvips:
SET TEXFONTS=%TEXTFM%

:: !!!! PostScript metafont mode, or printer resolution: !!!!
:: laserjet=300dpi; ljIV=600dpi
:: There should be a file \emtex\ps\%printer%.cfg and the Metafont
:: mode should be listed in \emtex\metafont\mfjob\modes.mfj
:: irrelevant for non-PostScript printing
SET PRINTER=laserjet
:: SET PRINTER=ljIV

:: MetaFont files (.MF)
SET MFINPUT=%EMTEXDIRSLASH%metafont\mfinput!

:: MetaFont base files
SET BMFBAS=%EMTEXDIRSLASH%METAFONT\BMFBASES

:: Metafont job files
SET MFJOB=%EMTEXDIRSLASH%METAFONT\MFJOB

:: MFJOB options
SET MFJOBOPT=/i /3

:: base directory for generated fonts
SET DVIDRVFONTS=C:\TEXFONTS

:: same for DVIPS:
SET TEXPKS=%DVIDRVFONTS%\%ddpi%\%f.%p;%EMTEXDIRSLASH%TEXFONTS%\printer%\%ddpi%\%f.%p

:: Virtual fonts (for dvips)
SET VFFONTS=.%EMTEXDIRSLASH%TEXFONTS\VF

:: directory for own font library ownfli.fli
:: the dvidrv cnf files use the environment variabele OWNFLI,
:: the dvips configuration file config.ps uses MYFONTS.
SET OWNFLI=%DVIDRVFONTS%
SET MYFONTS=%OWNFLI%

:: a few additional bells and whistles of the emTeX dvi drivers;
:: these environment variables may go undefined if dvi files
:: and graphic files are always in the current directory.
:: SET DRVDRVINPUT=.
:: SET DVIDRVGRAPH=.

:: some default options for the emtex dvi drivers
:: /fm: Metafont mode (see emtex\metafont\mfjob\modes.mfj)
:: These settings can be overruled on the dvidrv command-line.
SET DVIDRV=/fb /ft:1 /fj+ /fm:laserjet
SET DVIHPLJ=/po:lpt1
SET DVIDOT=/po:lpt1 /fm:epsonfx

:: DVIPS directory for configuration files; will add 'C:\' later
SET TEXCONFIG=%EMTEXDIRSLASH%PS

:: dvips printer-specific configuration file ::::::::::::::::::::
:: If you want to use your own PostScript fonts,
:: you can make them known to dvips in C:\MYPSPONTS.MAP

COPY %TEXCONFIG%\%PRINTER%.CFG C:\
ECHO M %PRINTER% >>C:\%PRINTER%.CFG
ECHO p +MYPSPONTS.MAP >>C:\%PRINTER%.CFG

SET TEXCONFIG=C:\;%TEXCONFIG%

:: dvips header files, eg. fonts; may add own directory
SET DVIPSHHEADERS=%EMTEXDIRSLASH%PS;%EMTEXDIRSLASH%PS\FONTS

:: Where GhostScript looks for files (apart from current directory).
:: This concerns initialization files, fonts and the file fontmap.
:: ***
SET GS_LIB=%EMTEXDIRSLASH%PS\FONTS;%EMTEXDIRSLASH%UTILS\GHOSTSCR;%EMTEXDIRSLASH%UTILS\SOURCE

:: DOSKEY MACRO'S ::::::::::::::::::::

:: NOTE. doskey syntax: '$*' means '%1 ... %9'

::: compiling :::

IF %LATEX%!=2E! doskey latex=%EMTEXDIRSLASH%compiler\tex386 -mt28000 &latex2e $*
IF %LATEX%!=209! doskey latex=%EMTEXDIRSLASH%compiler\tex386 -mt28000 &lplainn $*
IF %LATEX%!=209! doskey sliteX=%EMTEXDIRSLASH%compiler\tex386 -mt28000 &splainn $*

::: emTeX dvi drivers: previewing and printing :::
:: Note. '@hplj.cnf' calls a 'response file' hplj.cnf with options.
:: See emtex\data for additional configuration options.

:: previewing; automatic adapter selection
:: Windows users: be sure to also check out emtex\mwindow\dviwin
doskey tview=%EMTEXDIRSLASH%dvidrv %EMTEXDIRSLASH%dviscr /or+ /s3 @hplj.cnf @a4.pap $*

:: !!!! choose a tprint version or define your own !!!!!!!!!!!!!!!
:: You also may want to have a look at the batchfiles in \rawemtex
:: LaserJet <=3 and printing to <filename>.pcl
:: doskey tprint=%EMTEXDIRSLASH%dvidrv %EMTEXDIRSLASH%dvihplj @hplj.cnf /po=@Bi.pcl @a4.pap $*

```

```

:: LaserJet >=4 and printing to <filename>.pcl
doskey tprint=%EMTEXDIRSLASH%dvidrv %EMTEXDIRSLASH%dvihplj @ljiv.cnf /fm:ljiv /og600 /po=@Bi.pcl @a4.pap $*

:: DeskJet and printing to printer (use LaserJet driver)
:: doskey tprint=%EMTEXDIRSLASH%dvidrv %EMTEXDIRSLASH%dvihplj @hplj.cnf /op:deskjet @a4.pap $*

:: IBM Proprinter 4201: a non-pcl, non-PostScript printer using dvidot
:: doskey tprint=%EMTEXDIRSLASH%dvidrv %EMTEXDIRSLASH%dvidot ibm4201 @fx $*

::: PostScript dvi driver :::
:: Note. dvips writes default to <filename>.ps
:: A configuration file %printer%.cfg is read automatically.
doskey tps=%EMTEXDIRSLASH%dvidrv %EMTEXDIRSLASH%dvips32 $*

::: GhostScript: PostScript for non-PostScript devices :::
:: 'gs386 -?' shows available devices and the more important options

:: Viewing PostScript files. May replace vga with eg. s3vga or tseng.
:: Windows users: gswin.exe with or without gsvie.exe is preferable
:: but should be installed from the Windows Program Manager.
doskey gvview=%EMTEXDIRSLASH%UTILS\GHOSTSCR\gs386 -q -sDEVICE=vga $* quit.ps

:: Printing PostScript files to a non-PostScript printer

:: !!!! Replace ljet3 with eg. deskjet or djet500? !!!!!!!!!!!!!!!!
doskey gprint=%EMTEXDIRSLASH%UTILS\GHOSTSCR\gs386 -q -sDEVICE=ljet3 $* quit.ps

::: bibtex :::
doskey tbib=%EMTEXDIRSLASH%UTILS\bibtex32 $*

ECHO TeX configured
goto end

:: ERASING ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:ERASE
SET EMTEXDIR=0
SET EMTEXDIRSLASH=
SET RSX=
SET EMX=
SET LATEX=
SET TEXINPUT=
SET BIBINPUT=
SET BTEXFMT=
SET TEXTFM=
SET TEXFONTS=
SET PRINTER=
SET MFINPUT=
SET BMFBAS=
SET MFJOB=
SET MFJOBPT=
SET DVIDRVFONTS=
SET TEXPKS=
SET VFFONTS=
SET OWNFLI=
SET MYFONTS=
SET DRVDRVINPOT=
SET DVIDRVGRAPH=
SET DVIDRV=
SET DVIHPLJ=
SET TEXCONFIG=
SET DVIPSHEADERS=
SET GS_LIB=

IF EXIST C:\LJIV.CFG DEL C:\LJIV.CFG

DOSKEY LATEX=
DOSKEY SLITEX=
DOSKEY TEX=
DOSKEY TVIEW=
DOSKEY TPRINT=
DOSKEY TPS=
DOSKEY GVVIEW=
DOSKEY GPRINT=
DOSKEY TBIB=

echo TeX configuration erased
:end

```

Van dictaat tot boek

Cees Fortuin

Technische Faculteit
Hogeschool Gelderland
Ruitenberglaan 26
6826 CC Arnhem

Abstract

In 1980 werd ik docent natuurkunde aan de HTS te Arnhem voor de afdeling elektrotechniek. Toendertijd werd een leerboek gebruikt dat heel aardig de meer traditionele componenten behandelde, maar bijna niets wist te melden over wat tegenwoordig in een schakeling zit: dioden, transistoren en dat soort componenten. Een eerste navraag bij landelijke collega's leverde wel een soort werkgroepje op, maar niet de gezochte 'nieuwe' leerstof. Dus maar zelf schrijven. Dat moet in eerste instantie letterlijk worden opgevat, maar ook met een tekstverwerker als L^AT_EX moet veel zelf worden gedaan. Wat begon rond 1985 als dictaat, per hoofdstuk afgedrukt, eindigde rond 1995 als boek: twee delen 'Elektron natuurkunde' in de serie *natuurkunde voor het HBO* bij de Educatieve Uitgevers.

1 Van tekstverwerker naar tekstverwerker

In eerste opzet werd de geschreven tekst door een behulpzame hand bij de administratie uitgetikt: formules en tekeningen werden met de hand toegevoegd. In een volgend stadium werd de tekstverwerker van de Prime mini-computer gebruikt—maar dat was een ramp: praktisch onleesbare afdrukken; de studenten klaagden steen en been. De komst van de pc ging bij ons gepaard met 'Word Perfect'. Het lettertype werd daarbij ietsje beter, maar formules, in het bijzonder die in de lopende tekst, of griekse letters: ho maar. Door een artikel in, ik dacht, de American Mathematical Review, waarin tekstverwerkers werden vergeleken, kwam ik op het spoor van T_EX. Doch hoe aan een exemplaar te komen van een Amerikaans produkt: een collega die naar een 'vacantie' cursus aan het MIT ging kreeg een papier mee waarop mijn wensen stonden. Na (mijn) vakantie bleek hij een campusexemplaar op de kop te hebben getikt van μT_EX. Precies herinner ik me het niet meer, maar ik meen dat de fonts nogal onvolledig waren. Dus ging ik op zoek naar andere T_EX 'leveranciers', en die vond ik in eerste instantie bij de universiteit in Utrecht, waar PCT_EX werd gebruikt—overschakelen dus. Maar ook daar waren weer problemen: het bijgeleverde afdrukprogramma (is dat de vertaling van 'printerdriver'?) liet na zo'n 25 pagina's correct te hebben afgedrukt op de volgende lettertypen 'verdwijnen'. Ondertussen lid geworden van de NTG vond ik via de ledenlijst iemand die een betere 'dvidriver' wist: ik werd geruisloos fervent emT_EX gebruiker.

2 De decimale komma

Bij het schrijven kwamen verschillende problemen te voorschijn waarop niet een 'onmiddellijk' antwoord was. Sommige daarvan verdienen ruimere aandacht, zoals het pro-

bleem van de decimale komma. Van huis uit is T_EX een Amerikaanse tekstverwerker, het draagt daar de sporen van. In natuurkunde tekst komen natuurlijk heel veel grootheden voor, bestaande uit getallen en eenheden. In Amerika hebben getallen een decimale punt: 1.2, in Europa behoren zulke een decimale komma te hebben: 1,2; achter de komma wordt echter teveel ruimte gezet. Naast de *komma* en de *punt* heeft ook de *ster* een bijzondere positie binnen wiskundige tekst. Dus werden die drie aangepast met de volgende code (zie T_EXbook 344):

```
\gdef\plainmathcomma {\mathcode'\,=24891\relax}
%      reset plain mathcomma
\gdef\decimalekomma {\mathcode',=315 \relax}
%      ', ' als normale komma
\decimalekomma
%      initialisatie
\gdef\plainmathperiod {\mathcode'\.=314\relax}
%      reset plain mathperiod
\gdef\produktpunt {\mathcode'\.=513 \relax}
%      '.' als gewone cdot
\produktpunt
%      initialisatie
\gdef\plainmathstar {\mathcode'*=8707 \relax}
%      reset plain mathstar
\gdef\produktster {\mathcode'*=8706 \relax}
%      '**' als binaire maal
\produktster
%      initialisatie
```

Daarmee is 1,2 kilometer: 1,2·10³ m. Alles leek in orde, maar dat was het niet. In eerste aanzet waren de codegetallen namelijk *hexadecimaal* geschreven achter het symbool ". Helaas ging dat fout bij het gebruik van babel-nederlands: daar wordt " actief gemaakt. Overgaan naar *decimale* codegetallen was de eenvoudigste oplossing.

3 Het hok om formules

Ohm wist het al:

$$R = \frac{U}{I}$$

Om het belang van de formule aan te geven zetten we er een hok om. Maar `\[\fbox{...} \]` geeft helaas foutmeldingen vanwege missende `$`'s. Ook `\[\fbox{$. . . $} \]` geeft niet wat werd beoogd:

$$R = \frac{U}{I}$$

Ziet u wat er aan de hand is? De `\fbox` werkt alleen voor tekst (is dat nu een 'bug' van \LaTeX ?). Na enig gepuzzel werd de gebruikersvriendelijkere `\fbox`:

```
\let\LaTeXfbox=\fbox
\global\def\newfbox#1{\ifmmode\ifinner%
\LaTeXfbox{#1$}
\else\LaTeXfbox{\$ \displaystyle #1$}\fi
\else\LaTeXfbox{#1}\fi}
\let\fbox=\newfbox
```

waarmee Ohm's grootheid recht wordt gedaan (jawel, dit is dubbelzinnig taalgebruik):

$$R = \frac{U}{I}$$

4 Extensie bestanden

Een ander probleem deed zich voor bij het verwerken van antwoorden van opgaven en dergelijke. Met behulp van de \LaTeX 'environment' `list` was een `\opgave` gedefinieerd. Bij meerkeuzeopgaven worden de alternatieven gezet door een macro `\keus` of `\goedekeus`, die het antwoord (optioneel voorzien van commentaar) wegschrijft naar het bestand met extensie '.MKA'. Die antwoorden moesten per hoofdstuk worden verwerkt. Dat gaf twee problemen:

1. hde antwoorden (of de inhoudsopgave) *per hoofdstuk* af te drukken;
2. hoe het aantal gelijktijdig *openstaande* bestanden (aux, toc, lof, idx, mka, vwl, ...) te beperken;

\LaTeX maakt voor het gehele document één extensiebestand aan, alhoewel de .AUX bestanden per 'include'-bestand worden gemaakt. Gekozen werd voor een andere manier van omgaan met extensie bestanden.

Er werd een stijl: 'EXT.STY' ontworpen, die het verwerken van het '.AUX' bestand zodanig verandert dat ook extensie bestanden in principe per 'include'-bestand wordt verwerkt. Verder worden de verschillende extensiebestanden, aan het eind van het document, één voor één verwerkt, waardoor het aantal tegelijk openstaande bestanden drastisch verminderde. Daartoe moest `\enddocument` worden aangepast, hetwelk een forse versie afhankelijke ingreep is in de code. Verder waren aanpassingen nodig van `\include`, `\@starttoc`, `\@inputaux`, `\@writefile`, `\addtocontents`, `\addcontentsline`. Maar: nu bepaalt het bestand waarin de aanroep van bijv. `\tableofcontents` staat of alleen de inhoudsopgave van het hoofdstuk of die van alle hoofdstukken (of beide, indien gewenst) worden afgedrukt.

5 Figuren en tekst naast elkaar

Als iets me veel tijd heeft \LaTeX floats staan in een gekost, dan is het wel om 'vbox', die niet geknipt kan worden. Dus een simpele oplossing is: een `\vbox` waarin twee `verb+hbox` en naast elkaar staan. Helaas, verschillende problemen lagen op de loer: Hoe zorgen we ervoor dat de tekst in de 'box' precies aansluit (regelafstand) op de tekst boven en onder de 'box' (ook als die forse halen bevat)?

Een ander op te lossen probleem was hoe mee te gaan met de verschillende linker- en rechtermarges van de \LaTeX omgeving waarin de figuur wordt gezet. \LaTeX floats zijn (waren?) *ongevoelig* voor de omgeving waarin ze worden aangeroepen: een omissie naar mijn idee.

Net toen ik dacht alles netjes te hebben opgelost kwam ik voor nieuwe verrassingen te staan: de kop van een paragraaf stond opeens in de figuur (?). Zo leerde ik het speciale mechanisme van \LaTeX om kopjes te zetten kennen: via `\everypar`, die slechts bij de overgang van 'vmode' naar 'hmode' wordt aangeroepen. Ook dat werd opgelost; uiteindelijk kreeg de omgeving `\floatfix` gestalte. Ik ben nu tevreden over de manier waarmee figuren (tabellen) links of rechts (of pagina afhankelijk) van de tekst kunnen worden gezet, in aanpasbare breedte. De figuur zelf kan bijvoorbeeld als een `\picture`-bestand worden ingelezen. Onderschriften zijn 'bijna' \LaTeX -onderschriften. (Nou ja, een beetje aangepast wat de ruimtelijke instelling betreft, maar ik geloof dat de nieuwe $\text{\LaTeX}2_{\epsilon}$ daar nu ook in voorziet.)

6 Plaatjes: \Pictex en PS

Ter afsluiting de plaatjes. Binnen \LaTeX is er een zeer sterke ondersteuning van het zetten van wiskundige tekst zoals formules, maar wanneer het gaat om de visuele ondersteuning met grafieken en figuren is het heel wat minder. Dat lijnen slechts in een beperkt aantal richtingen worden ondersteund is een voorbeeld. Gelukkig ondersteunt \emTeX de `\emline` lijnen in willekeurige richtingen met *zijn* 'drivers'. Maar om andere lijnen te kunnen tekenen: *gestippeld* of *gestreept*, of om andere grafieken te kunnen tekenen als *ellipsen*, *hyperbolen*, *parabolen*, *wortelfuncties*... zijn andere hulpmiddelen nodig. Ik vond \Pictex , ook geschikt voor het *roteren* of *arceren* van figuren. Wel moesten de verschillende lengte-eenheden, lijndikten, puntgrootte, streep lengte en dergelijke van \LaTeX en \Pictex netjes worden afgestemd op elkaar. Hopelijk zullen bovenstaande 'wensen' allemaal in $\text{\LaTeX}3$ zijn ingebouwd, want nu was ik daar ook aardig wat tijd zoet mee. Misschien dat in $\text{\LaTeX}3$ alle 'rekenmachine functies' standaard beschikbaar kunnen zijn, zoals: `\plot{sin}`, `\plot{ln}`, `\plot{squ}`, `\plot{xtoy}`, `\plot{circular}`, ...?

Om tabellen dwars af te kunnen afdrukken (dat lijkt zo eenvoudig, maar is het niet) moest ik de hulp inroepen van PostScript. Ik gebruikte het ‘package’ ‘rotating.sty’ en het conversieprogramma DVIPS; ‘psboxit.sty’ was goed voor het zetten van rasters in figuren. Dan nog een laatste wens geformuleerd. Ik heb me afgevraagd waarom niet reeds in \TeX een optie is ingebouwd om fonts te kunnen spiegelen of dwars te draaien. Dat lijkt op het eerste gezicht niet echt moeilijk, en zou aan veel wensen tegemoet komen. Maar misschien heb ik problemen over het hoofd gezien (?).

7 Besluit

Voor mij was \LaTeX een onmisbaar hulpmiddel—natuurlijk, aangevuld met de vele ‘packages’ die door allerlei mensen zijn gemaakt en allerlei problemen oplossen—waarmee een dictaat kon worden omgezet in een ‘echt’ boek. Het ideaal: ‘de opmaak overlaten aan de tekstverwerker’ bleef echter bij een aantal onderwerpen buiten beeld, waardoor extra inspanning nodig was. Maar de opzet van \LaTeX kan dat ideaal wel dichterbij brengen. Bovendien: stoeien met problemen, ook bij tekstverwerking, vind ik persoonlijk nog leuk ook. Vandaar

Enkele ervaringen met L^AT_EX

Rom B.J. Pijlgroms

Afdeling Elektrotechniek en Informatica,
Technische en Maritieme Faculteit
Hogeschool van Amsterdam,
Europaboulevard 23,
1079 PC Amsterdam
pylgroms@htsa.hva.nl
<http://www.htsa.hva.nl/~pylgroms>

april 1996

Abstract

Ter sprake komen de persoonlijke ervaringen met L^AT_EX rond onder meer de uitwerkingen en antwoordenboeken van de methode ‘Wiskunde voor het hoger onderwijs’ (R. van Asselt, e.a., uitgave van: Educatieve Partners Nederland BV), de ervaringen met betrekking tot verzorgde bijscholingscursussen L^AT_EX t.b.v. het personeel van de faculteit en de overige tekstverwerkingsactiviteiten.

In het grijze verleden verzorgde ik ooit een cursus RUN-OFF (Prime-computer) ten behoeve van collega docenten. Bij deze tekstverwerker gaf je de opdrachten voor de tekststopmaak in de tekst zelf, maar wel telkens na een punt aan het begin van een nieuwe regel. Het principe van ‘code in tekst’ is de kern van wat er gebeurt in elk van de andere tekstverwerkingsprogramma’s die ik daarna leerde kennen. Hoe zou het ook anders moeten dan via integratie van tekst en code?

Dit idee is ook de basis van wat gebeurt in WordPerfect alhoewel dit voor menig gebruiker een geheim is; zelfs voor diegenen die de opmaakcode zichtbaar weten te maken via ALT-F3. Dat dit voor velen een geheim is komt doordat de user-interface het indrukken van (de meest ingewikkelde) toetscombinaties koppelt aan het plaatsen van codes in de tekst zelf. Het verborgen houden of zelfs volkomen verstoppen van de codes voor de gebruiker gaat nog een stap verder in de programma’s die werken volgens de Windows gebruikersinterface (WordPerfect voor Windows, MS-Word, enz.) Hier wordt het plaatsen van (verborgen) code gekoppeld aan het selecteren van tekstgedeelten en het indrukken van knoppen (en/of toetscombinaties). De slogan is: Klik, klak, klaar! Een extra winstpunt in deze omgeving is de What You See Is What You Get mogelijkheden (constant previewing).

In het licht van het bovenstaande zal het me niet verbazen als er nu (al) grafische user interfaces zijn die dezelfde kenmerken hebben ten aanzien van L^AT_EX. (In de WordPerfect formule-editor heeft men overigens de L^AT_EX-syntax ‘geleend’.)

Mijn ervaring met zowel het geven van cursussen aan docenten als het werken in een aantal auteursteams is dat relatief weinig personen de overstap wagen van traditionele tekstverwerker naar L^AT_EX. Zelfs als de te maken tekst een relatief grote formuledichtheid heeft verkiest men nog liever WordPerfect te gebruiken dan L^AT_EX ook maar te overwegen.

In het auteursteam van de methode ‘Wiskunde voor het hoger onderwijs’ heb ik een collega zover weten te krijgen dat we (vanaf de derde druk) samen de uitwerkingen- en antwoordenboeken als camera ready copy via L^AT_EX aandraagen. Voordat dit door de uitgeverij geaccepteerd werd heb ik zelf spitsroeden moeten lopen met het maken van een L^AT_EX-gezette versie van een uitwerkingen- en antwoordenboek bij een Statistiek boek. Voornaamste struikelblok bij de zeer kritische bureauredactie vormde de ‘rubber-witlengtes’ die L^AT_EX om de haverklap introduceert en waarover de gebruiker niet altijd iets te zeggen blijkt te hebben. De bureauredactie wilde een lay-out die mathematischer (strakker, mechanischer) was dan D. Knuth ons toestond. Heel wat tijd is gaan zitten in het verzinnen van oefjes om een en ander voor elkaar te krijgen. Uiteindelijk heeft de winst in productietijd (camera ready versus traditioneel zetten) de doorslag gegeven en kreeg de bureauredactie de opdracht meer door de vingers te zien.

Collega Wim Dolman (wim@htsa.hva.nl) heeft een aantal eigen styles geschreven, alsmede een eigen SliT_EX-omgeving om handouts (meer slides op een vel) voor studenten te maken. Speciale vermelding verdient een verbatim-style voor VHDL-files. Eén en ander is wel gedocumenteerd, maar wordt niet onderhouden; op aanvraag stelt hij het wel beschikbaar.

Een L^AT_EX-cursus in Groningen

Rein Smedinga

Abstract

Begin mei heeft bij de afdeling wiskunde en informatica van de Rijksuniversiteit Groningen voor de tiende keer een L^AT_EX-cursus plaatsgehad. Onderstaand een kort relaas over het ontstaan van deze cursus en het doel ervan.

1 Geschiedenis

Hoewel de vakgroep informatica vanaf het begin met L^AT_EX heeft gewerkt, prefereerde men bij de vakgroep wiskunde in Groningen in eerste instantie *Chi-writer*. Ruim vijf jaar terug wilde men toch over op L^AT_EX en gezien mijn ervaringen met L^AT_EX werd mij gevraagd de stafleden in enkele ochtenden de eerste beginselen bij te brengen.

Aangezien Theo Jurriëns (toen nog bij ruimte-onderzoek) al eerder L^AT_EX-cursussen had verzorgd (o.a. aan onze secretaresses), heb ik zijn materiaal genomen en verder uitgewerkt. Het belangrijkste aspect van de cursus was het zelf laten doen door de cursisten: van zelfdoen leer je tenslotte het meest. Uiteindelijk werd de cursus gegoten in een opeenvolging van steeds een kort stukje uitleg en een voorbeeld, gevolgd door één of meer opgaven over dezelfde stof.

De cursus had succes en iedereen was enthousiast over L^AT_EX.

2 Evolutie

Nadat de wiskundestaf de eerste beginselen van L^AT_EX was bijgebracht, kwamen ook steeds meer studenten, zowel van wiskunde als van informatica, bij mij met vragen over L^AT_EX. Zo ontstond het idee dezelfde cursus tweemaal per jaar aan te bieden aan wiskunde en informatica studenten (en nieuwe medewerkers binnen de afdeling). De cursus wordt nu twee keer per jaar gegeven: éénmaal na afloop van het eerste trimester (eind november) en éénmaal rond de paasvakantie (de laatste keer tijdens de meivakantie). In de loop van de tijd is de cursus hier en daar wat bijgeschaafd, maar in grote lijnen nog steeds zoals de eerste keer.

Aangezien vanuit het rekencentrum van de universiteit geen L^AT_EX-ondersteuning aanwezig is, is dit jaar besloten de cursus ook toegankelijk te maken voor studenten en medewerkers buiten de afdeling wiskunde en informatica.

3 Inhoud

De cursus bestaat uit vier dagdelen. Dag 1 is gewijd aan de eerste beginselen van het L^AT_EX-en. Bijzondere tekens, alsmede fonts en groepering, omgevingen en algemene documentindeling (zoals hoofdstukken, secties, voetnoten

en inhoudsopgave). Dag 2 behandelt voornamelijk tabellen en floats, de cross-referenties en geeft een eerste introductie in wiskundige formules. Dag 3 behandelt veel meer wiskunde, zoals array's, stelling-achtigen, en gaat verder in op het zelf definiëren van macros. De laatste dag gaat over documentpreparatie en behandelt naast de algemene indeling van een artikel en een rapport (o.a. met `\input` en `\include`) ook het maken van indexen (met `makeindex`) en referentielijsten (met `BiBTEX`).

Gedurende alle dagdelen wordt op dezelfde manier gewerkt: eerst een korte inleiding en behandeling van één of meer voorbeelden en vervolgens een aantal door de studenten te maken opdrachten. Dit herhaalt zich zo'n vier keer per dagdeel. Een aantal voorbeeld-files en templates zijn gemaakt en algemeen beschikbaar.

Groot succes is steeds het aan het eind verkrijgen van een diploma, waarbij de studenten *zelf* de oplossing van de in hun ogen moeilijkste opgave in dit diploma kunnen verwerken. Alleen de handtekening van de docent wordt nog handmatig aangebracht.

4 Toekomst

Aangezien wiskunde en informatica steeds meer met L^AT_EX werken is dit studiejaar besloten een deel van de cursus in de eerste week van de studie te geven. In deze zogenaamde introductieweek komen aan de orde: hoe werkt het afdelingspracticumsysteem (UNIX workstations), hoe maak je een verslag (L^AT_EX-cursus), hoe schrijf je een eenvoudig programma (cursus Pascal) en hoe doe je wiskundige berekeningen (cursus Mathematica).

Om Pascal-sources en mathematica-plaatjes in een verslag toe te kunnen voegen is de cursus uitgebreid met een uitleg over verbatim (L^AT_EX als typemachine) en over het invoegen van postscript-plaatjes.

Aangezien de eerste twee dagdelen van de L^AT_EX-cursus nu al in het eerste jaar van de wiskunde en informatica-studies zitten, wordt overwogen de eigenlijke L^AT_EX-cursus meer het karakter te geven van een cursus voor gevorderden. Behalve het wat meer uitdiepen van huidige onderwerpen in de cursus wordt overwogen extra onderwerpen toe te voegen, zoals het aanpassen van classes aan eigen wensen.

Het gebruik van L^AT_EX voor de wiskunde lessen

B.J. Geels *

Rinnewaard 45, 1824 JC Alkmaar

April 1996

Ongeveer acht jaar geleden kwam mij een werkstuk onder ogen dat wel erg mooi opgemaakt was. Bij navraag bleek dat gedaan te zijn met L^AT_EX. Daar ben ik toen direct achteraan gegaan, en met zeer veel moeite kreeg ik het systeem op onze XT-PC's op gang. Wat een gedoe en wat een geduld was er voor nodig.

Nu, een aantal handleidingen en boeken verder (en ook een paar computers verder) is mijn volharding zijn vruchten aan het afwerpen. Ik gebruik L^AT_EX momenteel voor al mijn proefwerken en mijn praktikumhandleidingen.

Tamelijk recent is daar een nieuwe impuls aan toegevoegd. Op de Vrije School wordt erg veel tijd besteed aan wiskunde lessen. Het is (net als alle andere vakken) geen keuzevak en moet dus op verschillende niveau's gegeven worden. In ons wiskunde programma worden andere onderwerpen behandeld en ook in andere volgorde dan in het staats- (op examen gerichte) programma. Al zijn verschillen gedurende afgelopen jaren minder geworden. De gangbare methoden konden wij zelfs als opgaven bundels maar met veel moeite gebruiken: steeds weer dingen veranderen, de dingen in andere leerjaren behandelen, dingen overslaan enzovoort. De nieuwste (alleen nog aangekondigde) veranderingen in het wiskunde programma zullen het nog moeilijker maken om opgaven uit deze boeken te kunnen gebruiken. Na veel experimenteren mijnerzijds blijkt het L^AT_EX systeem (en mijn kennis daarvan) zo stabiel en veelzijdig, dat we het wilden gaan gebruiken om eigen opgavenbundels te maken.

De belangrijkste stylen die we hiervoor gebruiken zijn de `exercise` en voor de figuren `picins` stylen. Verder is het zetten van opgaven enigszins vereenvoudigd door enkele macro's te schrijven en in een eigen `.sty`-file onder te brengen. Daarmee is onder andere de volgende opgave te zetten:

```
\begin{exercise}
Herleid:\\
\tweebijdrerij
{4\sqrt{2}-5\sqrt{2}-3\sqrt{2}}
\ldots
{-\frac{3}{2}\sqrt{5}-\derde\sqrt{5}}
\end{exercise}
```

met als resultaat:

1 Herleid:

$$\begin{array}{l|l} \text{a. } 4\sqrt{2} - 5\sqrt{2} - 3\sqrt{2} & \text{d. } -\sqrt{16} - 5\sqrt{16} - 4\sqrt{16} \\ \text{b. } -3\sqrt{7} - 4\sqrt{7} + 7\sqrt{7} & \text{e. } \frac{1}{2}\sqrt{10} + 2\frac{1}{2}\sqrt{5} \\ \text{c. } \frac{1}{3}\sqrt{5} + \frac{1}{6}\sqrt{5} & \text{f. } -\frac{3}{2}\sqrt{5} - \frac{1}{3}\sqrt{5} \end{array}$$

De macro's `\tweebijdrerij`, `\half` en `\derde` zijn in het genoemde aparte bestand gedefinieerd. Vooral tezamen met het gebruik van de optie `\answer` is dit zeer handig om snel goede opgaven reeksen met antwoorden te maken.

Een andere moeilijkheid was het probleem van de verschillende niveau's. Het is duidelijk dat men voor leerlingen die meer kunnen dan anderen, niet geheel nieuwe boekjes wil gaan schrijven. Men zou willen dat er een macro was die uit een reeks van opgaven, alleen die opgaven die met een bepaalde letter gemarkeerd zijn, selecteert, waarbij er opgaven voor meerdere niveau's gemarkeerd mogen zijn. Welnu de macro's die daarvoor zijn geschreven werkten niet goed samen met de optie `\answer` van de style `exercise`. Via het FGBBS zijn toen van Herman Haverkort goed werkende macro's ontvangen.

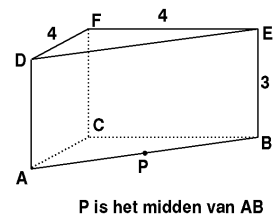
Ons rest nog een lastig probleem: tekeningen bij meetkunde opgaven.

2 Gegeven is het prisma

$ABCDEF$ en het punt P op AB .

a. Teken het snijpunt S van de lijn FP en het vlak BCD .

b. Het punt Q is het snijpunt van de lijnen AE en BD . Teken het snijpunt R van de lijn FQ en het vlak ABC .



P is het midden van AB

Sommige tekeningen kunnen met de `epic` of `eepic` opdrachten gezet worden maar de meeste tekeningen vragen iets meer. Mijn voorlopige oplossing is om ze aan te maken met WP-Presentations en dan als `.pcx`-plaatje te exporteren. Daarna kunnen ze met de volgende commando's naast de tekst gezet worden:

```
\parpic[r]{
\unitlength1cm
\begin{picture}(6,4)
\put(0,4){\special{em:graph .....pcx}}
\end{picture}}
```

*De auteur is leraar wiskunde, natuurkunde en computertechniek op de Adriaan Roland Holtschool, Vrije School voor voortgezet onderwijs te Bergen.

Van ms-troff naar L^AT_EX

Piet Tutelaers

P.T.H.Tutelaers@urc.tue.nl

In het begin van de 80-er jaren is door het Rekencentrum van de TUE een VAX-11/750 aangeschaft als centrale UNIX faciliteit voor medewerkers en studenten van de faculteiten. Een belangrijke toepassing was het pakket TROFF, in combinatie met EQN en TBL, voor het zetten van technische wetenschappelijke teksten. Speciaal voor dit doel hadden we een QMS1200¹ laserprinter aangeschaft. T_EX werd ons toen op advies van het CWI² afgeraden omdat het te ingewikkeld zou zijn in vergelijking met het ms-macro pakket in TROFF. Na enkele jaren een TROFF-cursus verzorgd te hebben, kwam ik toevallig in aanraking met het boek³ waarin Knuth op aanstekelijke wijze de ideeën achet zijn T_EX en METAFONT uiteenzet. Deze spraken me erg aan omdat TROFF, ontwikkeld uit runoff een versie voor niet-proportionele karakters, daarbij vergeleken toch wat achterbleef.⁴ Door het feit dat we gratis over een tape met T_EX, L^AT_EX en METAFONT konden beschikken, hebben we na enig experimenteren besloten van het MS-TROFF pakket over te stappen naar L^AT_EX.

De eerste L^AT_EX-cursus is gegeven in 1989 met PCT_EX waarvoor we een campuslicentie hadden aangeschaft. Het jaar 1993 is een belangrijk jaar voor T_EX gebruikers. Knuth implementeert immers 8-bits fonts met virtuele fonts zodat ook PostScript fonts kunnen worden gebruikt. Dankzij zijn EMX ontwikkelomgeving voor UNIX applicaties kon Eberhard Mattes vrij snel deze nieuwe T_EX versie overzetten naar MSDOS.⁵ Inmiddels wordt de L^AT_EX cursus één keer per semester gegeven met behulp van emT_EX aan studenten en medewerkers. De cursus behandelt in vijf ochtenden de inleiding uit het L^AT_EX boek van L^Ampport. We gaan ervan uit dat de cursisten ervaring hebben met MSDOS en met een editor kunnen omgaan op dat systeem. Wanneer DVIWIN virtuele fonts implementeert, het wachten is nog steeds op versie 3.0, dan stappen we over naar een MSWINDOWS versie van T_EX.

De cursus staat ook open voor mensen van buiten de TUE. Geïnteresseerden kunnen gratis onze cursusbrochure⁶ aanvragen of onze WWW pagina <http://www.tue.nl/urc/cursusraadplegen>.

¹Met een aanschafwaarde van meer dan een ton. De printer begreep nog geen PostScript maar QUICK

²Men beschikte hier over de TROFF goeroe Jaap Akkerhuis

³Knuth, Donald E., *T_EX and METAFONT: new directions in typesetting*. Providence: American Mathematical Society, 1979.

⁴Hierbij zal het feit dat de pionier van TROFF (Osanna) door een tragisch ongeval om het leven is gekomen, zeker hebben meegeespeeld

⁵Vele commerciële MSDOS T_EX versies moeten nog steeds virtuele fonts implementeren.

⁶Cursusadministratie Rekencentrum, Postbus 513, 5600 MB Eindhoven

The \LaTeX 2HTML translator: An Overview

Herbert W. Swan

dprhws@Arco.com

\LaTeX 2HTML is, as its name implies, a translator which converts a standard \LaTeX document into Hypertext Markup Language (*HTML*), for incorporation into the World-Wide Web. Like \LaTeX , it is freely available software, supported by highly dedicated volunteers. Unlike \LaTeX , it is currently available only on UNIX platforms. Complete documentation concerning this software may be browsed from <http://www-dsed.llnl.gov/files/programs/unix/latex2html/manual>. A European mirror is reported at <http://www.rzg.mpg.de/rzg/software/latex2html/>. From this online document, a PostScript version of the complete document may easily be obtained for hardcopy viewing. This article (the one you are reading) represents a condensation of that more complete documentation.

\LaTeX 2HTML replicates the basic structure of a \LaTeX document as a set of interconnected *HTML* files which can be explored using automatically generated navigation panels. The cross-references, citations, footnotes, the table-of-contents and the lists of figures and tables, are also translated into hypertext links. Formatting information which has equivalent “tags” in *HTML* (lists, quotes, paragraph-breaks, type-styles, etc.) is also converted appropriately. The remaining heavily formatted items such as mathematical equations, pictures or tables are converted to images which are placed automatically at the correct positions in the final *HTML* document.

\LaTeX 2HTML also extends \LaTeX by supporting arbitrary hypertext links and symbolic cross-references between evolving remote documents. It allows the specification of *conditional text* and the inclusion of raw *HTML* commands. These hyper-media extensions to \LaTeX are available as new commands and environments from within a \LaTeX document.

System Requirements

Before you consider obtaining \LaTeX 2HTML, you should first ensure that your UNIX system includes the following utilities:

- *Perl* version 4.036 or later. Since \LaTeX 2HTML is written in the *Perl* language, this utility is essential. Since *Perl* version 5 makes more efficient use of dynamic memory, it is recommended over version 4.
- Your UNIX system must support some form of keyed database management system, such as *DBM*, *NDBM*, *SDBM*, or *GDBM*. To find out whether your system supports one of these standards, do a “**man dbm**,” etc.

If you are using *SDBM* or *GDBM*, you must use *Perl* version 5.

- \LaTeX 2HTML can translate both \LaTeX 2 ϵ documents, as well as the older \LaTeX 2.09 documents. However, if you wish to use all of the features of the \LaTeX 2HTML translator, you should upgrade your \LaTeX program itself to the newer version 2 ϵ .
- If you want to be able to use \LaTeX to translate equations, figures and arbitrary environments into GIF images for web browsing, you will also need the following UNIX utilities available and working:
 - *dvips* version 5.516 or later or *dvipsk*. It is also recommended that your installation of *dvips* permits users to generate their own fonts through *Metafont* and the *MakeTeXPK* script. This will cause equations and other GIF text to be much more screen-readable.
 - *Ghostscript* version 2.6.1 or later, with the *ppmraw* device driver built in. (To see whether your version of *Ghostscript* contains this driver, type


```
gs
devicenames ==
quit
```
 - The *pbmplus* or *netpbm* library. Some of the image filters in these libraries are used during the PostScript to GIF image conversion (described below).
 - If you dislike the white background color of the generated inlined images then you should get either the *netpbm* library (instead of the older *pbmplus*) OR install the *giftrans* filter by Andreas Ley <ley@rz.uni-karlsruhe.de>. Version 1.10.2 is known to work without problems, but later versions should also be OK.

Obtaining \LaTeX 2HTML

One way \LaTeX 2HTML may be obtained is through one of the three primary Comprehensive \TeX Archive Network (CTAN) sites nearest you. They are located in the United States <ftp.shsu.edu>, the United Kingdom <ftp.tex.ac.uk>, and Germany <ftp.dante.de>. It can be found under the `tex-archive/support/latex2html` directory.

The CTAN version will always be the latest stable revision of \LaTeX 2HTML, currently V96.1-b. However, it will not always contain the latest patches submitted by the user community. If you wish to be on the leading edge,

you are better off obtaining the program from `<ftp://www-dsed.llnl.gov/files/programs/unix/latex2html/sources>` (European mirror `<ftp://ftp.rzg.mpg.de/pub/soft/latex2html/sources>`). This directory will contain all of the recent L^AT_EX2HTML releases, including the latest major release, 96.1, dated January of 1996. After downloading the file `latex2html-96.1.tar.gz`, you will have to unzip it with “`gzip -d`,” and extract the files with “`tar xvf latex2html-96.1.tar`.” You should then proceed to download the latest revision (currently Rev. e, dated April 8, 1996), `latex2html-96.1.reve.tar.gz`. Simply unzip and extract this file on top of the original release file for the latest patches. It will include `.diff` files which will show you what patches were made since the previous major release.

Installation

Once the L^AT_EX2HTML source files are obtained, installation consists of:

1. Editing the *Perl* scripts, to tell the system where the *perl* translator resides;
2. Editing the `latex2html.config` script to tell L^AT_EX2HTML where the necessary UNIX utility programs reside;
3. Running the `install-test` script, which mainly verifies the previous two steps;
4. Making local copies of the L^AT_EX2HTML icons so that they are accessible to your WWW server. The installation variable `$ICONSERVER` in `latex2html.config` must then be set to this location.

Warnings: If you cannot do that bear in mind that these icons will have to travel from Livermore, California!!! Also note that several more icons were added that were not present in previous versions of L^AT_EX2HTML.

5. Customizing the installation, as necessary. This is accomplished by setting various “installation variables” (described later) in the file `latex2html.config`. Individual users may also override the system-wide configuration, either globally for all their documents, for separately for individual documents.

For a “per user” initialization file, copy the file `dot.latex2html-init` in the home directory of any user that wants it, modify it according to her preferences and rename it as `.latex2html-init`. At runtime, both the `latex2html.config` file and `$HOME/.latex2html-init` file will be loaded, but the latter will take precedence.

You can also set up a “per directory” initialization file by copying a version of `.latex2html-init` in each directory you would like it to be effective. An initialization file `/X/Y/Z/.latex2html-init` will take precedence over all other initialization files if `/X/Y/Z` is the “current directory” when L^AT_EX2HTML is invoked.

Based on recent postings to the L^AT_EX2HTML mailing list (described below), the top three installation difficulties that new users encounter are:

Image conversion Although L^AT_EX2HTML is able to process ordinary text adequately, it fails to “find” the images to process. One common cause of this problem is that the *ppmraw* device is not built into *Ghostscript*. Another cause involves the `TEXINPUTS` environment variable. During normal operation, L^AT_EX2HTML creates a subdirectory under the current one to contain the translated *HTML* files. It is from this subdirectory that L^AT_EX and *dvips* are run during image conversion. If the `TEXINPUTS` environment variable that is visible to these programs does not include “..”, then image conversion will fail. At some sites, “*latex*” and “*dvips*” are really scripts which set `TEXINPUTS` and some other environment variables and call the actual programs. If this is the case, the user including “..” in his or her `TEXINPUTS` will not solve the problem.

Database problems Sometimes L^AT_EX2HTML will simply die if the *Perl* interface to the UNIX database routines is not working. In *Linux* systems, this problem is typically solved by uncommenting the “`use GDBM_File`” line in the L^AT_EX2HTML and `install-test` scripts. On other systems it may be necessary to recompile *Perl* with the proper database libraries.

File globbing failure This problem manifests itself by L^AT_EX2HTML creating only some temporary files, but no valid *HTML* files. This problem can be traced to the inability of *Perl* to locate the *cs*h program. Look for a line similar to “`cs`h= ‘*cs*h’” in the `Perl Config.pm` file. If it is absent, reconfigure *Perl*.

Normal Operation

To use L^AT_EX2HTML simply type: “`latex2html <file>.tex`.” The “`.tex`” suffix is optional and will be supplied by the program if omitted by the user. This will create a new directory called `<file>` which will contain the generated *HTML* files, some log files and possibly some images. To view the result use an *HTML* viewer such as NCSA Mosaic or Netscape Navigator on the main *HTML* file which is `file/file.html`. This file will contain navigation links to the other parts of the generated document.

The L^AT_EX2HTML script includes a short manual which can be viewed by typing “`nroff -man latex2html`.”

If a GIF image (of an equation or a figure, etc.) is used more than once in the document, it will be generated only once. Furthermore, on subsequent calls to L^AT_EX2HTML on the same document, most images need not be regenerated at all, but are recycled from one run to the next. Their names might change, but their contents do not. The only exception are images which are *order-sensitive*, meaning

that their content depends upon their location. Examples of order-sensitive images are `equation` and `eqnarray` environments. This is because their figure numbers are part of the image. Figures and tables with captions, on the other hand, are order-insensitive because the figure numbers are handled by *HTML* and are not part of the image itself.

For the most part, \LaTeX 2HTML acts as expected, processing *most* of the commands standard to \LaTeX . `\ref` and `\label` pairs are translated as anchor/hyper-references in the *HTML* document. \LaTeX lists are converted to *HTML* lists. Footnotes are collected onto one page and displayed as hypertext. Tables and certain inline equations are translated to their *HTML* equivalents, providing the selected *HTML* version supports these constructs.

Command-line options

It is possible to customize the output from \LaTeX 2HTML using a number of command-line options with which you can specify how to break up the document, where to put the generated files, what the title is, what the signature at the end of each page is, what to put in the navigation panel, what kind of extra information to include about the document, whether to retain the original \LaTeX section-numbering scheme, the version (below) of *HTML* to generate, etc.

The command-line options described below can be used to change the default behaviour of \LaTeX 2HTML. Alternatively the corresponding environment variables in the initialization file `.latex2html-init` may be changed, in order to achieve the same result.

- split** <num> Stop splitting sections into separate files at this depth. A value of 0 will put the document into a single *HTML* file. The default is 8.
- link** <num> Stop revealing child nodes at each node at this depth. (A node is a `\part` or `\chapter` or `\section` or `\subsection` or `\subsubsection` etc.). A value of 0 will show no links to child nodes, a value of 1 will show only the immediate descendents, etc. A value at least as big as that of the `-split` option will produce a table of contents for the tree structure, rooted at each given node. The default is 4.
- nolatex** Disable the mechanism for passing unknown environments to \LaTeX for processing. This can be thought of as “draft mode” which allows faster translation of the basic document structure and text, without fancy figures, equations or tables.
This option has been superseded by the `-no_images` option (see below).
- external_images** Instead of including any generated images inside the document, leave them outside the document and provide hypertext links to them.
- ascii_mode** Use only ascii characters and do not include any images in the final output. In ascii mode the output of the translator can be used on character-based browsers which do not support inlined images (the *HTMLIMG* tag).
- t** <top-page-title> Name the document using this title.

- up_url** <URL> Specifies a universal resource locator (URL) to associate with the “UP” button in the navigation panel(s).
- up_title** <string> Specifies a title associated with this URL.
- prev_url** <URL> Specifies a universal resource locator (URL) to associate with the “PREVIOUS” button in the navigation panel(s).
- prev_title** <string> Specifies a title associated with this URL.
- down_url** <URL> Specifies a URL for the “NEXT” button in the navigation panel(s).
- down_title** <string> Specifies a title associated with this URL.
- contents** <URL> Specifies a URL for the “TABLE OF CONTENTS” button, for document segments that would not otherwise have one.
- index** <URL> Specifies a URL for the “INDEX” button, for document segments that otherwise would not have an index.
- external_file** <filename-prefix> Specifies the prefix of the `.aux` file that this document segment should read. The `.aux` extension will be appended to this prefix. This file is necessary for document segments to obtain figure and section numbers from \LaTeX .
- dir** <output-directory> Redirect the output to this directory.
- no_subdir** Place the generated *HTML* files in the current directory. The default behaviour is to create (or reuse) another file directory.
- prefix** <filename-prefix> The <filename-prefix> will be prepended to all `.gif`, `.pl` and `.html` files produced, except for the top-level `.html` file. This will enable multiple products of \LaTeX 2HTML to peacefully coexist in the same directory. However, do **not** attempt to *simultaneously* run multiple instances of \LaTeX 2HTML using the same output directory, else various temporary files will overwrite each other.
- font_size** <size> This option provides better control over the font size of environments processed by \LaTeX . <size> must be one of the font sizes that \LaTeX recognizes; i.e. 10pt, 11pt, 12pt, etc. The default size is the same as the \LaTeX document. Whatever size is selected, it will be magnified by the installation variables `$MATH_SCALE_FACTOR` or `$FIGURE_SCALE_FACTOR`.
- no_tex_defs** If specified, the translator will make no attempt to interpret raw \TeX commands. This feature will enable sophisticated authors the ability to insert arbitrary \TeX commands in environments that are destined to be processed by \LaTeX anyway; e.g. figures, theorems, etc.
- ps_images** Use links to external PostScript images rather than inlined GIF images.
- address** <author-address> Sign each page with this address.
- no_navigation** Disable the mechanism for putting navigation links in each page.

- top_navigation** Put navigation links at the top of each page.
- bottom_navigation** Put navigation links at the bottom of each page as well as the top.
- auto_navigation** Put navigation links at the top of each page. Also put one at the bottom of the page if the page exceeds `\$WORDS_IN_PAGE` number of words (default = 450).
- index_in_navigation** Put a link to the *index-page* in the navigation panel if there is an index.
- contents_in_navigation** Put a link to the *table-of-contents* in the navigation panel if there is one.
- next_page_in_navigation** Put a link to the *next logical page* in the navigation panel.
- previous_page_in_navigation** Put a link to the *previous logical page* in the navigation panel.
- info** `<string>` Generate a new section *About this document...* containing information about the document being translated. The default is to generate such a section with information on the original document, the date, the user and the translator. An empty string (or the value 0) disables the creation of this extra section. If a non-empty string is given, it will be placed in the contents of the *About this document...* section instead of the default information.
- reuse** `<reuse.option>` The `<reuse.option>` specifies how or whether image files are to be shared or recycled, and accepts three valid options:
 - 0 Do not ever share or recycle image files. This choice also invokes an interactive session prompting the user about what to do about a pre-existing *HTML* directory, if it exists.
 - 1 Recycle image files from a previous run if they are available, but do not share identical images that must be created in this run.
 - 2 Recycle image files from a previous run and share identical images from this run. (This is the default).
- no_reuse** Do **not** share or recycle images generated during previous translations. This is equivalent to `-reuse 0`. (This will enable the initial interactive session during which the user is asked whether to reuse the old directory, delete its contents or quit.)
- init_file** `<file>` Load the specified file. This *Perl* file will be loaded after loading `\$HOME/.latex2html-init`, if one exists. It can be used to change the default options.
- no_images** Do not attempt to produce any inlined images. The missing images can be generated “off-line” by restarting $\text{\LaTeX}2\text{HTML}$ with the option `-images_only`.
- images_only** Try and convert any inlined images that were left over from previous runs of $\text{\LaTeX}2\text{HTML}$.
- show_section_numbers** Show section numbers. By default the section numbers are not shown in order to encourage the use of particular sections as stand-alone documents. In order to be shown, section ti-

cles must be unique and must not contain inlined graphics.

- html_version** `<version>` This specifies the *HTML* version (See below). to generate. Currently, versions 2.0, 2.1, 2.2, 3.0 and 3.1 are available. The default version is 2.0.
- vs** Print the current version of $\text{\LaTeX}2\text{HTML}$.
- debug** Run in debug mode, using the *Perl* debugger, if it is installed.
- h** Print out the list of options.

HTML version

The Hypertext Mark-up Language is an evolving standard, with different versions supporting different features. In order to make your documents viewable by the widest possible audience, you should use the most basic *HTML* version in common usage. This is currently version 2.0 which is the default version for $\text{\LaTeX}2\text{HTML}$. It supports images, server-side image maps, interactive forms, and minimal typographic elements (bold, italic and teletype). This version adheres to the ISO–Latin–1 (ISO–8879) character set. Other *HTML* versions supported by $\text{\LaTeX}2\text{HTML}$ are:

Version 2.1 *HTML* version 2.1 is essentially identical to *HTML* version 2.0, with some extensions for internationalization. Most importantly, the default character set is no longer ISO–8859–1 but ISO–10646, commonly known as *Unicode*. This is a 16-bit character set and can thus display a much larger set of characters.

Version 2.2 This version supports all the features of version 2.1, plus the *HTML3* Table Model. This feature is already available in many *HTML* browsers, including Netscape Navigator V1.2 and later.

Version 3.0 This version adds to version 2.2 some of the *HTML* 3.0 textual formatting features, including centering, flush right, flush left and underlining.

Version 3.1 $\text{\LaTeX}2\text{HTML}$ revision “d” and later support *HTML* subscripts and superscripts, which may be browsed by Navigator V2 and later.

Version 3.2 In addition to all of the features of version 3.1, this version adds support for math mark-up. Currently the only browsers which can display this mark-up are *Arena* (<http://www.w3.org/hypertext/www/Arena>) and *UdiWWW for Windows 95* (<http://www.uni-ulm.de/~richter/udiwww/>)

Very few native \TeX commands are supported. The ones that are include: `\newdimen`, `\newbox`, `\vskip`, `\hskip`, `\char`, and some limited forms of `\def`. Furthermore, `\def` and `\renewcommand` may not behave as expected, if they define the same entity more than once in the same document. Furthermore, macro definitions cannot contain any verbatim-like environments.

Internationalization

A special variable `$LANGUAGE_TITLES` in the initialization or configuration files determines the language in which some section titles will appear. For example setting it to `$LANGUAGE_TITLES = ``french``` will cause \LaTeX 2HTML to produce “Table des matières” instead of “Table of Contents”. Furthermore, the value of `\3rd June 1996` is presented in a format which is customary in that language.

The only languages currently supported are `french`, `english` and `german`, but it is trivial to add support for another language in the file `latex2html.config`. As a guide here is the entry for the French titles:

```
sub french_titles {
  $toc_title = "Table des mati\\`eres";
  $lof_title = "Liste des figures";
  $lot_title = "Liste des tableaux";
  $idx_title = "Index";
  $bib_title = "R\\`ef\\`erences";
  $info_title =
    "\\`A propos de ce document...";
  $abs_title = "R\u00e9sum\u00e9";
  $pre_title = "Pr\u00e9face";
  $app_title = "Annexe";
  @Month = ( 'janvier', 'f\u00e9vrier',
             'mars', 'avril', 'mai',
             'juin', 'juillet', 'ao\u00fbt',
             'septembre', 'octobre',
             'novembre', 'd\u00e9cembre' );
}
```

In order to provide full support for another language you may also want to replace the navigation buttons which come with \LaTeX 2HTML (which are by default in English) with your own. As long as the new buttons have the same filenames as the old ones, there should not be a problem.

Hypertext Extensions to \LaTeX

One of the greatest appeals of the World-Wide Web is its high connectivity through hyperlinks. \LaTeX 2HTML implements several \LaTeX extensions designed to rapidly exploit this connectivity. These extensions are made known to \LaTeX by using the `html` package (style file) included with the \LaTeX 2HTML distribution.

The simplest (but most tedious) way to insert a hypertext link into your document is by use of the `\htmladdnormallink` command. This command takes two arguments: the text to highlighted as hypertext, and the URL that that hypertext will take you if selected. Images may be inserted in your document via `\htmladdimg`, which takes as its single argument the URL of the image to be inserted. These commands are time-consuming for the author, because exact URLs must be supplied, and because they must be manually updated if the destination URLs are ever changed.

Part of the author’s burden of maintaining hyperlinks is relieved by the use of symbolic links. The \LaTeX `\label` and `\ref` mechanism is one way to providing such links. If the target link is another \LaTeX 2HTML-processed document, this mechanism may be extended by:

1. Substituting the `\externalref` command for the `\ref` command, and
2. Including an `\externallabels` command somewhere in the document to may a correspondence between the URL of the target document and its \LaTeX 2HTML table of symbols. Such a symbol table is automatically generated by \LaTeX 2HTML for every document.

Still another way of cross referencing text, either within or between documents, is with the `\htmlref` command. This macro is similar to `\htmladdnormallink`, except that the second argument is replaced by a symbolic \LaTeX label, defined by `\label`. If the label not found within the current document, it may instead appear in any of the external \LaTeX documents listed in an `\externallabels` command. There is also a `\hyperref` construction which produces a different wording of the link in the printed version of the document than in the electronic one.

If a portion of the text is to appear only in the *HTML* version of the document, it may be enclosed within the `htmlonly` environment. Short text segments destined only for *HTML* may alternately be placed in the `\html` command. Conversely, portions of the document intended only for the printed version may be enclosed within the `latexonly` environment, or the `\latex` command.

Any other *HTML* construct may be inserted by way of the `rawhtml` environment. Any text contained within this environment is passed directly to the *HTML* file without modification. This text could include forms, frames, or any other *HTML* object.

Navigation panels

The navigation panels are the strip containing “buttons” and text that appears at the top and perhaps at the bottom of each generated page and provides hypertext links to other sections of a document. Some of the options and variables that control whether and where it should appear have already been mentioned.

A simple mechanism for appending customized buttons to the navigation panel is provided by the command `\htmladdtonavigation`. This takes one argument which \LaTeX 2HTML appends to the navigation panel. For example,

```
\htmladdtonavigation{\htmladdnormallink
  {\htmladdimg{http://server/URL}}{%
  http://server/link}}
```

will add an active button `mybutton.gif` pointing to the specified location.

Apart from these facilities it is also possible to specify completely what appears in the navigation panels and in what order. As each section is processed, \LaTeX 2HTML assigns relevant information to a number of global variables. These variables are used by the subroutines `top_navigation_panel` and `bottom_navigation_panel`, where the navigation

panel is constructed as a string consisting of these variables and some formatting information.

These subroutines can be redefined in a system or user configuration file (respectively, `LATEX2HTMLDIR/latex2html.config` and `\$HOME/.latex2html-init`). Any combination of text, *HTML* tags, and the variables mentioned below is acceptable.

The control-panel variables are:

Iconic links (buttons):

PREVIOUS Points to the previous section;
UP Points up to the “parent” section;
NEXT Points to the next section;
NEXT_GROUP Points to the next “group” section;
PREVIOUS_GROUP Points to the previous “group” section;
CONTENTS Points to the contents page if there is one;
INDEX Points to the index page if there is one.

Textual links (section titles):

PREVIOUS_TITLE Points to the previous section;
UP_TITLE Points up to the “parent” section;
NEXT_TITLE Points to the next section;
NEXT_GROUP_TITLE Points to the next “group” section;
PREVIOUS_GROUP_TITLE Points to the previous “group” section.

If the corresponding section exists, each iconic button will contain an active link to that section. If the corresponding section does not exist, the button will be inactive. If the section corresponding to a textual link does not exist then the link will be empty.

The number of words that appears in each textual link is controlled by the variable `$WORDS_IN_NAVIGATION_PANEL_TITLES` which may also be changed in the configuration files.

```
sub top_navigation_panel {
  # Start with a horizontal rule (3-d dividing line)
  "<HR> ".
  # Now add few buttons with a space between them
  "$NEXT $UP $PREVIOUS $CONTENTS $INDEX $CUSTOM_BUTTONS" .
  # Line break
  "<BR>\n" .
  # If ``next`` section exists, add its title to the navigation panel
  ($NEXT_TITLE ? "<B> Next:</B> $NEXT_TITLE\n" : undef) .
  # Similarly with the ``up`` title ...
  ($UP_TITLE ? "<B>Up:</B> $UP_TITLE\n" : undef) .
  # ... and the ``previous`` title
  ($PREVIOUS_TITLE ? "<B> Previous:</B> $PREVIOUS_TITLE\n" : undef) .
  # Horizontal rule (3-d dividing line) and new paragraph
  "<HR> <P>\n"
}
```

Figure 1: Sample Perl subroutine to be inserted in the \LaTeX 2HTML configuration or startup file to customize the document’s top navigation panel

Figure 1 shows an example of a navigation panel specification. (The “.” is the *Perl* string concatenation operator and “#” signifies a comment).

Converted images

\LaTeX 2HTML copies the contents certain commands and all environments that it was not specifically programmed to handle into the file `images.tex` for GIF file conversion. This conversion is done by way of \LaTeX , *dvips*, *Ghostscript*, and the portable bitmap library. The contents of these commands and environments are not altered by \LaTeX 2HTML, except for macro expansion, but this can be defeated if necessary. The complete \LaTeX preamble from the user’s document is also passed to `images.tex`, so that user macros, packages and graphics options (like `\graphicspath`) are made available to \LaTeX processing the images. The *environments* which are processed as images include:

- Inline equations, excepts those which can be processed by \LaTeX 2HTML according to its *HTML* version, specified as above;
- `equation` and `eqnarray`, except when running under *HTML* version 3.2.
- `table` and `tabular`, except under *HTML* version 2.2 or higher;
- `figure`, `floatfig`, `wrapfig`, and other floating-figure environments;
- `theorem` and `minipage` environments;
- Any user-defined environment (as described below) specified to \LaTeX 2HTML that should be passed to \LaTeX .

The standard \LaTeX *commands* which are converted to images include `\fbox`, `\framebox`, `\parbox`, `\dag`, `\ddag`, `\oe`, `\OE`, and certain special accents not defined in *HTML*. Also converted to images are the `graphics` package commands, `\rotatebox`, `\scalebox`, `\reflectbox`, `\resizebox`, `\includegraphics`, `\epsfig`, `\psfig`, `\epsffile`, and `\epsfbox`.

On the other hand, the following text color commands are processed as ordinary text by L^AT_EX2HTML, if they occur outside of a command or environment that would otherwise be passed to L^AT_EX for image conversion: `\color`, `\textcolor`, `\pagecolor`, `\colorbox`, `\fcolorbox`.

Macro interactions

If a L^AT_EX graphics package requires one or more T_EX definitions to be made within each graphics environment (e.g. `eepic`), there are currently several ways to proceed:

1. If it is possible to define the required macros only once, do that at the top of the document. L^AT_EX2HTML will expand each reference to the macros, before passing them on to L^AT_EX for image conversion.
2. If the T_EX macros need to *change* with each figure, it is not possible to make them within each figure. L^AT_EX2HTML will expand all references to the macros according to their *last* definition in the document. Instead, place the entire figure, macros and all, into a separate file, say `chart.eepic`. Then in your main document, insert `"\fbox{\InputIfFileExists{chart.-eepic}}."` The command `\InputIfFileExists` is not recognized by L^AT_EX2HTML, and macros contained within it will be left untouched. It *is* recognized by L^AT_EX, however, and will be processed by it correctly.
3. Another method of having multiple, variable macro definitions is to make them with native T_EX `\def` commands, and to use the `-no_tex_defs` command-line switch. With this switch, T_EX `\def`'s are not recognized by L^AT_EX2HTML, and will be left unexpanded as they are copied to `images.tex`.

Scaling and orientation

Images created by this mechanism can be divided into two categories: "small" and "large." For purposes of discussion ...

"**small images**" refers to equations, special accents and any other image generating L^AT_EX commands; while ...

"**figures**" applies to any image-generating L^AT_EX environments (e.g. `figure`, `table`, `minipage` etc.)

The size of all "small images" depends on a configuration variable `MATH_SCALE_FACTOR` which specifies how much to enlarge or reduce them in relation to their original size in the PostScript version of the document. For example a scale factor of 0.5 will make all images half as big while a scale factor of 2 will make them twice as big. Larger scale factors result in longer processing times and larger intermediate image files. A scale factor will only be effective if it is greater than 0. The configuration variable `FIGURE_SCALE_FACTOR` performs a similar function for "figures". Both of these configuration variables are initially set to 1.6.

For finer control, several parameters affecting the conversion of a single "figure" can be controlled with the command `\htmlimage`, which is defined in `html.sty`. The one argument of `\htmlimage` is a string of options separated by commas. The options are:

- `scale=` <scale factor>
- `external`
- `thumbnail=` <scale factor>
- `map=` <server-side image map URL>
- `usemap=` <client-side image map URL>
- `flip=` <flip_option>
- `align=` <alignment>

The `scale=` option allows control over the size of the final image. It overrides the global `FIGURE_SCALE_FACTOR` for this one figure.

The `external` option will cause the image not to be inlined. (Images are inlined by default.) External images will be accessible via a hypertext link.

The `thumbnail=` option will cause a small inlined image to be placed in the caption. The size of the thumbnail depends on the <scale factor>. Use of the `thumbnail=` option implies the `external` option.

The `map=` and `usemap=` options specify that the image is to be made into an active image map (See below).

The `flip=` option specifies a change of orientation of the electronic image relative to the printed version. The <flip_option> is any single command recognized by `pnmfip`. The most useful of these include:

rotate90 or r90 This will rotate the image clockwise by 90°.

rotate270 or r270 This will rotate the image counter-clockwise by 90°.

leftright This will flip the image around a vertical axis of rotation.

topbottom This will flip the image around a horizontal axis of rotation.

The `align=` option specifies how the figure will be aligned. The choices are: `top`, `bottom`, `middle`, `left`, `right` and `center`. The `middle` option specifies that the image is to be left-justified in the line, but centred vertically. The `center` option specifies that it should also be centred horizontally. This option is valid only if the `HTML` version is 3 or higher, or if the `NET_SCAPE_HTML` configuration variable is set. The default value is `bottom`.

In order to be effective the command `\htmlimage` and its options must be placed inside the environment on which it will operate.

Active Image Maps

Image maps are images with active regions in which a Web surfer can click, to send him off to another sector of cyberspace. L^AT_EX2HTML can design either inline "figures" or external ones (with or without a thumbnail version) to be image maps. However, `HTML` requires a URL of an `HTML map-file` which defines the coor-

dinates of each active region in the map with a destination URL. Usually this map file is kept on the server machine, however *HTML* version 3 also allows it to reside on the client side for faster response. (See `<http://ds.internic.net/internet-drafts/draft-seidman-clientsideimagemap-02.txt>` Both configurations are supported by $\text{\LaTeX}2\text{HTML}$ through the `\htmlimage` options “`map=`” and “`usemap=`,” respectively.

Keeping such a map file up to date manually can be tedious, especially with dynamic documents under revision. An experimental program *makemap* can help automate this process. This program (which is really a *Perl* script) takes one mandatory argument and one optional argument. The mandatory argument is the name of a *user-map* file, defined below. The optional argument is the name of the directory where the *HTML* map file(s) are to be placed.

The best way of describing how this works is by example. Suppose that a document has two figures designated to become active image maps. The first figure included a statement like,

```
\begin{figure}
\htmlimage{map=/cgi-bin/imagemap/Blk.map}
. . .
\end{figure}
```

The second figure had a line like,

```
\begin{figure}
\htmlimage{map=/cgi-bin/imagemap/Flow.map}
. . .
\end{figure}
```

A typical user map file, named `report.map`, might contain the following information¹:

```
+report/ URL
#
# Define map #1:
#
Blk.map:
label1 rect 288,145 397,189
label2 rect 307,225 377,252
label2 default
#
# Define map #2
#
Flow.map:
label3 circle 150,100 200,100
label4 default
```

In this file, comments are denoted by a #-sign in column 1. The line beginning with `+report` states that the symbolic labels are to be found in the `labels.pl` contained in the directory `report/`, and that its associated URL is as stated. Any number of external `labels.pl` files may be so specified. The block diagram image has two active regions. The first is a rectangle bounded by corners (288, 145) and (397, 189), while the second is a rectangle bounded by corners (307, 225) and (377, 252). These

coordinates can be obtained with the aid of a program such as *xv*. If the user clicks in the first rectangle, it will cause a branch to the URL associated with symbolic label `label1` defined in the `labels.pl` file found in directory `report/`. The single active region in the flow chart figure is a circle centred at (150, 100) and passing through point (200, 100). Clicking in this region will cause a branch to symbolic label `label3`. Labels `label2` and `label4` will be visited if the user clicks anywhere outside of the explicit regions. If any labels are not defined in any of the `labels.pl` files mentioned, they will be interpreted as URLs without translation.

The *HTML* image maps are generated and placed in directory `report/` by invoking the command: “**makemap report.map report.**”

Fancy lists

An optional style file `htmllist.sty` has been provided which produces fancier lists in the electronic version of the document. This file defines a new \LaTeX environment, `htmllist`, which causes a user-defined item-mark to be placed at each new item of the list, and which causes the optional description to be displayed in bold letters. The mark is determined by the `\htmlitemmark { <item-mark> }` command. This command accepts either a mnemonic name for the `<item-mark>`, from a list of icons established at installation, or the URL of a mark not in the installation list. The command `\htmlitemmark` must be used inside the `htmllist` environment in order to be effective, but it may be used more than once to change the mark within the list. The item-marks supplied with $\text{\LaTeX}2\text{HTML}$ are `BlueBall`, `RedBall`, `OrangeBall`, `GreenBall`, `PinkBall`, `PurpleBall`, `WhiteBall` and `YellowBall`. The `htmllist` environment is identical to the description environment in the printed version.

An example of its usage is:

```
\begin{htmllist}
\htmlitemmark{WhiteBall}%
\item[Item 1:] This will have a
white ball.
\item[Item 2:] This will also have a
white ball.
\htmlitemmark{RedBall}%
\item[Item 3:] This will have a red ball.
\end{htmllist}
```

Change bars

$\text{\LaTeX}2\text{HTML}$ supports the $\text{\LaTeX}2\epsilon$ `changebar.sty` package, written by Johannes Braams `<JLBraams@cistron.nl>`, for inserting *change-bars* in a document in order to indicate differences from previous versions. Changed text is surrounded by bracket GIF icons.

¹This file is designed for an NCSA server. CERN servers use “`rect`” instead of “`rectangle`,” specify a radius instead of an outer point in the circle, and enclose point coordinates by parentheses.

Document Segmentation

As we have seen, the \LaTeX author can provide these links either manually or symbolically. Manual links are more tedious because a URL must be provided by the author for every link, and updated every time the target documents change. Symbolic links are more convenient, because the translator keeps track of the URLs. Earlier releases of \LaTeX 2HTML required the entire document to be processed together if it was to be linked symbolically. However, it was easy for large documents to overwhelm the memory capacities of moderate sized computers. Furthermore, processing time could become prohibitively high, if even a small change required the entire document to be reprocessed.

For these reasons, program segmentation was developed. This feature enables the author to subdivide his document into multiple *segments*. Each segment can be processed independently by \LaTeX 2HTML. Hypertext links between segments can be made symbolically, with references shared through auxiliary files. If a single segment changes, only that segment needs to be reprocessed (unless a label is changed that another segment requires). Furthermore, the entire document can be processed without modification by \LaTeX to obtain the printed version. The top level segment that \LaTeX reads is called the *parent* segment. The others are called *child* segments.

Program segmentation does require a little more work on the part of the author, who will now have to undertake some of bookkeeping formerly performed exclusively by \LaTeX 2HTML. The following four \LaTeX extensions carry out segmentation:

```
\segment { <file> } { <sec-type> } { <heading> }
```

This command indicates the start of a new program segment. The segment resides in `<file>.tex`, represents the start of a new \LaTeX sectional unit of type `<sec-type>` (e.g., `\section`, `\chapter`, etc.) and has a heading of `<heading>`. A variation of this command, `\segment*`, is provided for segments that are not to appear in the table of contents. These commands perform the following operations in \LaTeX :

1. The specified sectioning command is executed.
2. \LaTeX will write its section and equation counters into an auxiliary file, named `<file>.ptr`. It will also write an `\htmlhead` command to this file. This information will tell \LaTeX 2HTML how to initialize itself for the new document segment.
3. \LaTeX will then proceed to input and process the file `<file>.tex`.

The `\segment` and `\segment*` commands are ignored by \LaTeX 2HTML.

```
\internal[ <file> ] { <prefix> }
```

This command directs \LaTeX 2HTML to load intersegment information of type `<type>` from the file `<prefix><type>.pl`. Each program segment must be associated with a unique filename-prefix, specified either through a command-line option, or through

the installation variable `$AUTO_PREFIX`. The information `<type>` must be one of the following:

internals This is the default type, which need not be given. It specifies that the internal labels from the designated segment are to be input and made available to the current segment.

contents The table of contents information from designated segment are to be made available to the current segment.

sections Sectioning information is to be read in. Note that the segment containing the table of contents requires both contents and sections information from all other program segments.

figure Lists of figures from other segments are to be read.

table Lists of tables from other segments are to be read.

index Index information from other segments are to be read.

```
\startdocument The \begin{document} and
\end{document} statements are contained in the
parent segment only. It follows that the child seg-
ments cannot be processed separately by \LaTeX, with-
out modification. However they can be processed sep-
arately by \LaTeX2HTML, provided it is told where the
end of the \LaTeX preamble is; this is the function of
the \startdocument directive. It substitutes for
\begin{document} in child segments, but is oth-
erwise ignored by both \LaTeX and \LaTeX2HTML.
```

```
\htmlhead{ <sec-type> } { <heading> } This com-
mand is not normally placed in the document at all.
It is automatically passed from parent to child via
<file>.ptr. It identifies to \LaTeX2HTML that the
current segment is a \LaTeX sectional unit of type <sec-
type>, with the specified heading. This command is
ignored by \LaTeX itself.
```

A segmentation example

The best way to illustrate document segmentation is through a simple example. Suppose that a document is to be segmented into one parent and two child segments. Let the parent segment be `report.tex`, and the two child segments be `sec1.tex`, and `sec2.tex`. The latter are translated with filename prefixes of `s1` and `s2`, respectively. This example is included with the version 96.1 distribution of \LaTeX 2HTML, with more prolific comments than are shown here.

The text of `report.tex` is given below:

```
\documentclass{article}
\usepackage{html,makeidx,color}
```

```
\internal[figure]{s1}
\internal[figure]{s2}
\internal[sections]{s1}
\internal[sections]{s2}
\internal[contents]{s1}
\internal[contents]{s2}
```

```

\internal[index]{s1}
\internal[index]{s2}

\begin{document}
\title{A Segmentation Example}
\date{\today}
\maketitle
\tableofcontents
\listoffigures

% Process the child segments:

\segment{sec1}{section}{Section 1 title}
\segment{sec2}{section}{Section 2 title}
\printindex
\end{document}

```

This file obtains the information necessary to build an index, a table of contents and a list of figures from the child segments. It then proceeds and typesets these.

The first child segment, `sec1.tex`, is shown below:

```

\begin{htmlonly}
\documentclass{article}
\usepackage{html,color,makeidx}
\input{sec1.ptr}
\end{htmlonly}
\internal{s2}
\startdocument
Here is some text.
\subsection{First subsection}
Here is subsection
  1\label{first}.
\begin{figure}
\colorbox{red}{Some red
  text\index{Color text}}
\caption[List of figure
  caption]{Figure 1 caption}
\end{figure}
Reference\index{Reference} to
  \ref{second}.

```

The first thing this child segment does is establish the \LaTeX packages it requires, then loads the counter information that was written by the `\segment` command that invoked it. Since this segment contains a symbolic reference (`second`) to the second segment, it must load the internal labels from that segment.

The final segment, `sec2.tex`, is shown below:

```

\begin{htmlonly}
\documentclass{article}
\usepackage{html,makeidx}
\input{sec2.ptr}
\end{htmlonly}
\internal{s1}
\startdocument
Here is another section\label{second}.
Plus another\index{Reference, another}
reference\ref{first}.
\begin{figure}
\fbox{The figure}
\caption{The caption}
\end{figure}

```

This segment needs to load internal labels from the first one, because of the reference to `first`. These circular dependencies (two segments referencing each other) are either not allowed or handled incorrectly by the Unix utility `make`, without resorting to time stamps and some trickery. A *time-stamp* is a zero-length file whose only purpose is to record its creation time. Besides evaluating segment interdependence, another function of `make` is to provide inter-segment navigation information.

A sample Makefile is included in the distribution. This correctly generates the fully-linked document. The first time it is invoked, it runs:

- \LaTeX on `report.tex` twice;
- `dvips` to generate `report.ps`;
- \LaTeX 2HTML on `sec1.tex`;
- \LaTeX 2HTML on `sec2.tex`. At this point, `sec2.html` is completely linked, since the labels from the `sec1` were available;
- \LaTeX 2HTML on `sec1.tex` to pick up the labels from `sec2`;
- \LaTeX 2HTML on `report.tex`.

Proper operation of `make` depends on the fact that \LaTeX 2HTML updates its own internal label file only if something in its current program segment causes the labels to change from the previous run. This ensures that \LaTeX 2HTML is not run unnecessarily. It is also usual for the information page to be suppressed by specifying `-info` on all but the top-level program segment.

Extending \LaTeX 2HTML

As the translator covers only partially the set of \LaTeX commands and because new \LaTeX commands can be defined arbitrarily using low level TeX commands, the translator should be flexible enough to allow end-users to specify how they want particular commands to be translated.

Adding Support for Specific Style Files

\LaTeX 2HTML provides a mechanism where code to translate specific style files is automatically loaded, if such code is available. When use of a style, such as `german.sty`, is detected in a \LaTeX source document, the translator looks for a file `LATEX2HTMLDIR/styles/german.perl`. If one is found, then it will be loaded into the main script.

This mechanism will help to keep the core script smaller as well as make it easier for others to contribute and share solutions on how to translate specific style files. The current distribution includes the files listed in Table 1. These will provide good examples of how you can create your own extensions to \LaTeX 2HTML.

.perl file	Description
alltt	Supports the \LaTeX 2 ϵ 's alltt package.
changebar	Provides rudimentary change-bar support.
color	Causes colored text to be processed as ordinary text by \LaTeX 2HTML.
french	Special support for the French language.
epsfig	Processes embedded figures not enclosed in a figure environment.
floatfig	Processes floating figures.
german	Special support for the German language.
graphics	Supports commands in the graphics package.
graphicx	Supports the alternate syntax of graphics commands.
heqn	Alters the way displayed equations are processed.
htmlist	Provides support for fancy lists.
makeidx	Generates the index.
texdefs	Supports some raw \TeX commands.
wrapfig	Supports wrapped figures.

Table 1: Supported \LaTeX 2HTML style files.

The problem however, is that writing such extensions requires an understanding of *Perl* and of the way \LaTeX 2HTML is organised. Interfaces that are more “user-friendly” will be investigated.

At the moment a rudimentary mechanism is provided so that a user can ask for particular commands and their arguments either to be ignored or passed on to \LaTeX for processing (the default behaviour for unrecognized commands is for their arguments to remain in the *HTML* text). Commands that are passed on to \LaTeX are converted to images which are either “inlined” in the main document or become accessible via hypertext links. Simple extensions using the commands above may be included in the $\$LATEX2HTMLDIR/latex2html.config$ file or in each personal $\$HOME/.latex2html-init$ initialization file.

Adding ignored Commands

Commands that should be ignored may be specified in the $.latex2html-init$ file as input to the `ignore_commands` subroutine. Each command which is to be ignored should be on a separate line followed by compulsory or optional argument markers separated by `#`'s e.g.²:

```
<cmd_name>#{}# []# {}# [] ...
```

²It is possible to add arbitrary *Perl* code between any of the argument markers which will be executed when the command is processed. For this however a basic understanding of how the translator works and of course *Perl* is required.

`{}`'s mark compulsory arguments and `[]`'s optional ones.

Some commands may have arguments which should be left as text even though the command should be ignored (e.g. `\mbox`, `\` etc.). In these cases arguments should be left unspecified.

Here is an example of how this mechanism may be used:

```
&ignore_commands( <<_IGNORED_CMDS_);
documentstyle # [] # {}
linebreak# []
center
<add your commands here>
_IGNORED_CMDS_
```

Passing Commands to \LaTeX

Commands that should be passed on to \LaTeX for processing because there is no direct translation to HTML may be specified in the $.latex2html-init$ file as input to the `process_commands_in_tex` subroutine. The format is the same as that for specifying commands to be ignored. Here is an example:

```
&process_commands_in_tex
(<<_RAW_ARG_CMDS_);
fbox # {}
framebox # [] # [] # {}
<add your commands here>
_RAW_ARG_CMDS_
```

New Features

A number of new features were introduced in the 96.1 version of \LaTeX 2HTML. These changes were provided by the users of this program. Some of these have already been discussed. The more significant ones include the following:

Font generation: The quality of equation bitmaps is greatly improved by enabling the `PK_GENERATION` configuration variable. When this is done *Metafont* will be invoked through `dvips` to generate fonts more suitable to screen viewing. Ideally, this should be done by setting the `mode=` switch to `dvips`, but unfortunately not all versions of `dvips` support this option. For those that don't, a `.dvipsrc` file is supplied with this distribution.

Active image-maps: Both server and client-side maps are supported. Image-maps can either be inline or external. External maps can be associated with a thumbnail image. A separate script `makemap` helps resolve external URLs in image-maps.

Improved image recycling: The older version of image recycling often caused images to overwrite each other due to confusion in the bookkeeping. This has been fixed. It is also no longer necessary to keep generating the same image which is being used repeatedly in a document. Furthermore, images with thumbnails can now be recycled, as well as active image-maps. Only images of the correct size are recycled.

Document segmentation: Large documents can now be divided into independently processed segments. Additional command-line switches provide inter-segment navigation information, while automatically generated *Perl* files pass symbolic references and counter information between segments.

Command parsing: Constructions such as `\hello2` are now treated as macro `\hello` followed by argument '2', rather than as macro `\hello2`. Added `\makeatletter` and `\makeatother` commands.

Graceful termination: When L^AT_EX2HTML is interrupted by a termination signal, all child tasks are also terminated. This provides a more orderly shut-down than that offered by previous versions.

More inline-math: Small inline equations which can be typeset in *HTML* are typeset in *HTML*. This further reduces the number of GIF files that need to be generated.

External hypertext: Commands `\html\ref` and `\hyper\ref` now accept labels defined in an external document, if the internal reference is not found.

Additional style files:

htmlist Defines a fancier list environment which embellishes each item of a description list with a user-selected icon. (It is the same as the `description` environment in the paper version.)

heqn Redefines the `equation` environment so that equation numbers are handled in *HTML*. This causes equations in this environment to be recyclable. It also causes equation arrays to be recyclable if their equation numbers do not change from the previous run.

floatfig Provides support for this environment by making it look like an ordinary figure in the electronic version.

wrapfig (Same comment as for `floatfig`).

graphics Defines elements of the standard L^AT_EX2_ε `graphics.sty` package.

L^AT_EX2_ε support: Provided support for the `\ensuremath` command. The last *option* to the `babel` package is interpreted as the L^AT_EX2HTML language style-file to load. User-defined commands and environments can now have an optional argument. Stubs have been provided for `\enlargethispage` and `\suppressfloats`. L^AT_EX2_ε packages `alltt`, `graphics`, `graphicx`, `color`, `changebar` and `epsfig` were provided.

Figure orientation: The `flip=` option of `\html-image` causes the program `pnmflip` to be called prior to the generation of the GIF file. This allows the electronic version of the image to be oriented differently than the paper version.

Null images: If for some reason an image produced a null GIF file, then no reference is made to that file and the program proceeds gracefully. Furthermore, such null images do not need to be regenerated. This would occur in *HTML3.0* images whose caption is enclosed in a `\parbox`, for example.

Navigation panels: There are now separate subroutines to control the top and bottom navigation panels.

Section headings: Labels, equations and images are now permitted inside a section heading, and in the `\title` command. However, the equations may not look very good because of the size difference.

Future directions

Work is currently underway on a complete rewrite of L^AT_EX2HTML. The new version will be based much more on T_EX's grammar and parsing. According to Marcus E. Hennecke, doctoral student at Stanford University, the rewrite (probably to be named V96.3) "does away with any kind of preprocessing, thus eliminating *texexpand*, *DBM* files, etc. and instead works directly on the raw L^AT_EX code. I would imagine that it should be possible to run that version on a PC except for the image generation, which will be the only thing requiring external helper programs." One minor drawback of V96.3 is that it will *require* users to upgrade to *Perl 5*. For this reason, versions based on V96.1 will continue to be supported in parallel for some time to come.

Hennecke emphasizes that L^AT_EX2HTML V96.3 is not, by itself, a port to non-UNIX platforms. However, it should make such a port easier to achieve, since it will store all system dependencies in variables, rather hard-coded into the source. Anyone contemplating working on a port from UNIX should corroborate with him directly at marcush@crc.ricoh.com.

User Support

To keep in touch with other users of L^AT_EX2HTML, to get additional assistance, or to suggest or provide bug fixes, you can join the online mailing list, provided through the Argonne National Labs. To subscribe to this list, send a message to: latex2html-request@mcs.anl.gov with the contents: **subscribe**. To be removed from the list send a message to the same address with the contents: **unsubscribe**. All recent postings to this discussion group are archived in a web-browsable at <http://www.rosat.mpe-garching.mpg.de/mailling-lists/LaTeX2HTML/>. You may not get immediate answers to all your question, but most inquiries are answered eventually. Furthermore, comments, suggestions and critiques are all helpful for the the continuing development of this evolving program.

³<http://cbl.leeds.ac.uk/nikos/tex2html/>

Acknowledgements

$\text{\LaTeX}2\text{HTML}$ was originally written by Nikos Drakos³ while he was with the Computer Based Learning Unit of the University of Leeds. The material in this article was drawn heavily on the $\text{\LaTeX}2\text{HTML}$ user's documenta-

tion, as revised most recently by myself, Ross Moore⁴ and Michel Goossens⁵. Thanks also go to Donald Arseneau⁶ for his `url.sty` \LaTeX package, which was instrumental in formatting this article.

⁴email: <ross@mpce.mq.edu.au>

⁵email: <goossens@cern.ch>

⁶email: <asnd@trumf.ca>

Zijn Scientific Word en T_EX uitwisselbaar?

Jan Jacobs and Jan-Egbert Sturm
Faculteit der Economische Wetenschappen
Rijksuniversiteit Groningen*

Eens in de zoveel tijd steekt de discussie de kop op over welke tekstverwerker het beste is. De nieuwste loot aan de stam wat dat betreft is Scientific World (SW). Het ligt voor de hand SW met T_EX te vergelijken, immers SW is een Windows-shell om T_EX.

Ten principale huldigen we het standpunt dat iedereen vooral moet blijven werken met het tekstverwerkersprogramma waarmee hij/zij het meest vertrouwd is. Overgaan op een nieuwe versie van een tekstverwerker of een andere tekstverwerker werkt in ieder geval op korte termijn contra-productief, het wekt een hoop frustratie op, en in het gunstigste geval worden de geproduceerde teksten fraaiër doch zeker niet beter.

Over het algemeen is SW aan te bevelen voor mensen die nog niet eerder met T_EX hebben gewerkt, maar die graag fraaiere uitvoer willen hebben, vooral als er veel formules worden gebruikt. SW is minder geschikt voor degenen die maximale vrijheid en uitwisselbaarheid eisen. Maar voor ons is maximale vrijheid geen issue. Het zou moeten gaan om de inhoud van een verhaal en veel minder om hoe het er uit ziet. Schrijven vervalt heel vaak tot window-dressing: ander lettertype, andere bladspiegel, gekleurde plaatjes? Misschien zouden we dat soort dingen toch moeten overlaten aan mensen die er verstand van hebben. Met moderne tekstverwerkers is het gemakkelijk te doen, je raakt eraan gewend en je raakt verwend. Desalniettemin zouden enige beperkingen op zijn plaats zijn: maximale vrijheid is iets voor tekstmakers, niet iets voor tekstschrijvers.

Waar we wel veel belang aan hechten is uitwisselbaarheid. Samenwerking stelt eisen aan het gebruik van tekstverwerkers. Een WP-er kan niet overweg met T_EX-files, en omgekeerd kent een T_EX-er slechts één WP commando: F7 of hoe kom ik hier zo snel mogelijk uit. Het gezamenlijk produceren van teksten betekent van tevoren een keuze maken over de tekstverwerker, hetgeen kan impliceren dat er nog wel eens dubbel werk wordt gedaan. Gelukkig valt dat vaak wel mee. De meeste tekstverwerkersprogramma's kunnen moeiteloos stukken ASCII inlezen. Het valt echter niet mee een artikel te editen met een tekstverwerker waar je niet vertrouwd mee bent.

In dit stuk zullen we daarom ingaan op de uitwisselbaarheid tussen SW en T_EX. Bij het laatste hebben we niet de 4T_EX workbench op het oog, maar de configuratiebestanden van Siep Kroonenberg¹ waarmee wij op Erik Frambach's [4]T_EX installatie werken. Deze worden vanaf 1992 gebruikt door een aantal medewerkers van het Centrum van Conjunctuur en StructuurOnderzoek (CCSO). Zonder problemen mogen we stellen, met uitzondering van een paar korte perioden waarin de directory-structuur van de T_EX-installatie moest worden aangepast aan de recentste ontwikkelingen.

Wij werken aan een artikel voor een conferentie. De eerste auteur van dit stuk gebruikt T_EX, de tweede SW. Met een paar kleine aanpassingen blijkt het heel goed mogelijk met twee verschillende tekstverwerkers aan hetzelfde artikel te werken.

Waar aan te denken?

In de eerste plaats, statements die bij SW boven het `\begin{document}` statement staan worden opgevat als stijl opties. In een L^AT_EX stuk mogen de titel van het stuk, de naam van de auteurs en zelfs de samenvatting voor het `\begin{document}` staan. Met `\maketitle` komen ze toch wel op de goede plaats. Je moet dus beginnen met `\begin{document}` voor de titel enz. te zetten.

Ten tweede, een klein punt wat we direct hebben aangepakt is dat SW standaard uitgaat van L^AT_EX_{2 ϵ} , terwijl we in onze T_EX-configuratie de conversie nog niet hebben gemaakt. SW kan echter gemakkelijk worden ingesteld op L^AT_EX 2.09.

Ten derde, de stijlen lijken een probleem. SW heeft – zoals iedere TeX configuratie – zijn eigen stijlen-bibliotheek. We hebben hiervoor een ad-hoc oplossing geïmplementeerd. De stijlen die we in het T_EX-document aanroepen (bijvoorbeeld `array` en `dcolumn`) hebben we in de SW directory gezet, en zijn daardoor in staat onze stukken er (bijna) net zo mooi uit te laten zien als vroeger. Onze ervaring is dat de meeste T_EX-stijlen probleemloos werken in SW. Of het omgekeerde ook het geval is hebben we niet getest. Eén van de weinige stijlen waarmee wij problemen hebben ondervonden – en die we niet met een (ad-hoc) op-

*Eerder in een iets andere vorm verschenen in EZ, het personeelsblad van de Faculteit der Economische Wetenschappen der Rijksuniversiteit Groningen

¹Zie Siep Kroonenberg, *T_EX without 4T_EX on the PC* in dit nummer

lossing hebben weten te verhelpen – is het aanroepen van de `mathtime`-stijl die het lettertype `mathtimes` aanroept, waarmee dezelfde fontfamilie wordt gebruikt voor tekst en formules. Een beetje \TeX -goeroe moet hier wat ons betreft snel een acceptabele oplossing voor kunnen vinden. Dit geldt overigens ook voor plaatjes. In onze implementatie verslikt SW zich niet in de stijl `psfig`, omdat we ook deze in de SW-directory hebben geplaatst. Het is ons echter nog niet gelukt om de plaatjes ook daadwerkelijk uit te printen. Wij hebben ondertussen vernomen dat het aan de praat krijgen van `mathtime` en `psfig` niet triviaal is.

Ten vierde, een onhebbelijkheid van SW is het `\text` commando waarmee platte tekst in een formule kan worden gezet. \TeX -ers gebruiken hiervoor het `\mbox` commando. Het is echter niet al te moeilijk om een \LaTeX -macro `\text` te definiëren zodat TeX deze optie direct aankan. Overigens, de stijlen van de American Mathematical Society gebruiken dezelfde conventie voor `\text`.

Ten slotte

Samenvattend, wat betreft het soort teksten die wij produceren zijn SW en \TeX uitwisselbaar zolang rekening wordt gehouden met elkaars onhebbelijkheden. Door alleen de echt noodzakelijke stijlen aan te roepen wordt hier al gauw aan voldaan. Verder dienen alle stijlen die worden gebruikt in de SW directory te staan. Blijken stijlen toch niet in beide pakketten te werken, dan kunnen zowel SW-notatie als \TeX -notatie worden opgenomen. Door vervolgens de niet relevante notatie uit te commenten kan alsnog in beide pakketten met het document worden gewerkt.

We verwachten dat het niet lang meer zal duren dat \TeX -ers, mits ze zijn uitgerust met de krachtige machines die SW vraagt, zullen overstappen op SW, al is het alleen maar om gebruik te maken van de edit-faciliteiten van deze Windows toepassing. Ze kunnen namelijk vrij gemakkelijk van alle voordelen van de standaard \TeX -configuratie gebruik blijven maken.

Typografisch Programmeren

J. Hagen

PRAGMA, Onderwijskundig Bureau voor Advies- en Ontwikkelwerk,
Postbus 125, 8000 AC Zwolle, (038) 422 9775

Abstract

In dit artikel wordt de ontwikkeling van een interactief document beschreven. De getoonde voorbeelden demonstrenen de kracht van T_EX als typografische programmeertaal. Omdat dergelijke documenten het moeten opnemen tegen (vaak dedicated) programma's, kan het ontwikkelen ervan worden omschreven als *typografisch programmeren*.

1 Inleiding

Met de komst van (1) snelle computers, (2) opslagmedia met grote capaciteit en (3) viewers als Acrobat Reader dient zich voor T_EX een nieuw werkterrein aan: de productie van complexe, interactieve teksten. In dit artikel gaan we in op enkele aspecten van het produceren van zulke teksten.

De afdeling Control van de Koninklijke Landmacht is verantwoordelijk voor de distributie van documenten binnen dit krijgsmachtdeel. We praten in dat verband al snel over duizenden documenten, variërend van een partituur voor het Wilhelmus (2 varianten) tot beschrijvingen van een Leopard-tank. Omdat het begrip publicatie in de ruimste betekenis van het woord wordt opgevat, komen we ook stickers, formulieren en sjablonen tegen.

Elk jaar worden er door de betreffende afdeling catalogi uitgegeven van publicaties. Bovendien worden in de loop van het jaar updates rondgestuurd. Aangezien het aantal publicaties in de duizenden loopt en het aantal abonnee's op de catalogi rond de tweeduizend ligt, zal duidelijk zijn dat de jaarlijkse verschijning ervan aardig wat papier kost.

Tot voor kort werden de catalogi samengesteld uit prints van de database waarin de gegevens zijn opgeslagen. Zowel de omvang in termen van papier als de wens volledige updates te verstrekken, zijn aanleiding geweest elektronische uitgave te overwegen. De beschikbaarheid van Acrobat Reader op verschillende platforms en de zich aandienende standaard formaat PDF voor elektronische documenten, opende de weg daartoe. Overigens werken op dit moment de meeste gebruikers met MSDOS.

In dit artikel gaan we in op de wijze waarop twee catalogi met behulp van het in huis ontwikkelde macropakket CONTEX_T zijn geproduceerd. We doen dit aan de hand van voorbeelden van de schermen zoals de gebruiker die ook te zien krijgt.

2 Het eerste document

We beginnen met de eerste catalogus: die van afzonderlijke publicaties.

Bij wijze van introductie krijgt de gebruiker een titelblad te zien. Aan de rechterkant van het scherm zien we het navigatiegedeelte, links staat de tekst. De aan te klikken delen hebben een wat zwaardere achtergrond en de woorden cq. karakters worden weergegeven in kleur. Zowel achtergronden als aanvullende ruimte voor navigatie-instrumenten worden standaard ondersteund door CONTEX_T.

In de toelichting vertellen we uit welke databases het document is opgebouwd. Het is de ruwe uitvoer van deze bestanden die aan de basis staat van het document. Alle gegevens van een publicatie stonden daarbij op één regel.



In eerste instantie lieten we T_EX de verschillende velden uit de regel filteren. Dit gebeurde op basis van een serie database-macro's, waarbij de velden op naam konden worden opgeroepen. Bovendien was een mechanisme ingebouwd om gecodeerde velden om te zetten. Tegenover het voordeel van onafhankelijkheid stond het nadeel van lange rekentijden. Dit soort handelingen zijn nu eenmaal niet T_EX's sterkste kant. Bovendien gaven bijzondere en actieve karakters problemen. (Hier zouden enkele e-T_EX opties wellicht uitkomst hebben geboden.)

Uiteindelijk is besloten de aangeleverde bestanden om te zetten in een wat sneller te behappen formaat. Het hiertoe geschreven programma zal te zijner tijd worden uitgebreid, maar kon in dit verband worden beperkt tot het (met behulp

van een definitiefile) genereren van een file met commando's als:

```
\beginofrecord
\field{name 1}{data of field 1}
\field{name 2}{data of field 2}
\endofrecord
```

TOELICHTING		publicatie
Deze verzamel-index KL-INDEX/EP is samengesteld en uitgegeven onder verantwoordelijkheid van de sectie Documentatie- en Publikatiebeleid (DPB) van de directie Control.		categorie
Het is een bundeling van de volgende binnen de KL van kracht zijnde indexen:		trefwoord
VS 2-200	Materieel-Logistieke Publikaties KL	
	Verzorgingspublicaties	
VS 2-2505	Formulieren	helpinfo
VS 2-2506	Boeken	colofon
Deze elektronische publikatie wordt uitgegeven in het Portable Document Format (PDF). Voor de distributie wordt gebruik gemaakt van een digitale gegevensdrager.		

Bij het verwerken van dergelijke records en velden wordt veelvuldig gebruik gemaakt van `\curname`, maar dat terzijde.

Het verwerken van het eerste bestand, dat uit bijna 3000 records bestaat, kost als we afzien van interactiviteit ongeveer een half uur. Het verwerken van de definitieve interactieve versie duurt drie maal zo lang. Deze verwerkingstijden zijn mede een gevolg van de vrij uitgebreide functionaliteit van `CONTEXT`. Hier betalen we de prijs van flexibiliteit.

HELP INFORMATIE (1)		publicatie
DEBKLEP start onder WINDOWS full-screen op. De full-screen-mode wordt verlaten met Esc, waarbij de menu's van Acrobat Reader beschikbaar komen. DEBKLEP start onder MSDOS niet full-screen op. Voor de mogelijkheden van Acrobat Reader wordt verwezen naar de Help in de menubalk.		categorie
Toegang tot de gewenste informatie wordt verkregen met behulp van aanklikbare, groen gekleurde woorden en pijltjes. Aanklikken met de muis resulteert in een sprong in het document.		trefwoord
menubuttons	publicatie naar de lijsten met publicaties categorie naar het register met categorieën trefwoord naar het register met trefwoorden toelichting naar de toelichting op dit document helpinfo naar de helpinformatie bij dit document colofon naar het colofon	toelichting
toetsen	PageUp naar het vorige scherm PageDown naar het volgende scherm Home naar het eerste scherm van het document End naar het laatste scherm van het document Esc verlaat de full-screen-mode	colofon

Bij de helpinformatie zien we naast de buttons voor heen- en weerbladeren een nieuw navigatie-instrument opduiken. De twee blokjes staan voor de twee pagina's waaruit de helpinformatie bestaat. Deze blokjes worden automatisch gegenereerd en maken gebruik van het in `CONTEXT` aanwezige mechanisme om subpagina's te nummeren.

Het subpaginanummer is zichtbaar gemaakt in de titel. Deze titel wordt herhaald omdat, in tegenstelling tot een papieren document, in een interactief document de tactiele en visuele signalen ontbreken met betrekking tot de plaats in het document. Bovendien kunnen we minder informatie

kwijt op een bladzijde. Hoewel een soort `\everypage` denkbaar is, is hier gebruik gemaakt van kopregels (headers). Omdat een kopregel meerdere regels tekst kan bevatten, wordt per hoofdstuk de kopregel opnieuw ingesteld.

HELP INFORMATIE (2)		publicatie
publicaties	naar de sublijst met publicaties naar de publicatie	categorie
categorieën	naar de eerste publicatie in deze categorie naar de middelste publicatie in deze categorie naar de laatste publicatie in deze categorie	trefwoord
trefwoorden	naar de eerste publicatie met dit trefwoord naar de middelste publicatie met dit trefwoord naar de laatste publicatie met dit trefwoord	toelichting
categorieveld	naar de vorige publicatie in deze categorie terug naar de lijst met categorieën naar de volgende publicatie in deze categorie	helpinfo
trefwoordveld	naar de vorige publicatie met dit trefwoord terug naar de lijst met trefwoorden naar de volgende publicatie met dit trefwoord	colofon
navigatiebuttons	naar het eerste scherm van het document naar het vorige scherm naar het volgende scherm naar het laatste scherm van het document	

COLOFON		publicatie
Redactie	Directie Control Landmachistaf Interne Zaken Sectie Documentatie- en Publikatiebeleid MPC 58 A Postbus 90821 2509 LV Den Haag tel: 06-546 65274 fax: 06-546 64228	categorie
Vormgeving	PRAGMA, Onderwijskundig Bureau voor Advies- en Ontwikkelwerk in Zwolle.	trefwoord
Productie	CDP Grafische Bedrijven in Emmen in samenwerking met DMKL, LIV, LIS en Bureau Informatiebeheer.	toelichting
Datum	De vaststeldatum van deze publicatie is 30-01-1996. De gebruikte gegevens zijn afkomstig uit de informatiesystemen CDOOS en CPUVOOS en geven de situatie van 30-01-1996 weer.	helpinfo
Bestelwijze	De KL-INDEX/EP is aan te vragen met het bijgevoegde formulier DF 1. Uw aanvraag kunt u indienen bij uw Beheerder Boekwerken, die voor de verdere afhandeling zal zorgdragen.	

Naast deze introducerende bladzijden bevat het document eigenlijk nauwelijks typografisch aantrekkelijke informatie. Dit is mede het gevolg van de beperkingen van databases. Bovendien wordt inconsistent gebruik van spaties, leestekens en haakjes, door `TEX` genadeloos afgestraft in de vorm van overfull boxes. Vandaar dat in dit document het afbrekmechanisme soepel is ingesteld. Bovendien worden enkele velden (door `TEX`) op orde gebracht. Zo zetten we afgebroken woorden als `zie-zo` om in `ziezo`. Dergelijke afbrekingen komen voor omdat in de database de informatie over meerdere regels kan worden ingevoerd.

De publicaties zijn ondergebracht in ongeveer vijftig categorieën. Daarnaast is aan iedere publicatie een trefwoord gekoppeld en heeft elke publicatie een uniek nummer.

De eerste ingang in het document loopt via het publicatienummer. Via een categorie komt men bij de lijsten met publicaties per categorie. Deze lijsten worden automatisch gegenereerd.

Binnen `CONTEXT` kan men in principe een onbeperkt aantal lijsten definiëren. Standaard zijn lijsten beschikbaar met hoofdstukken, paragrafen, subparagrafen enz. Derge-

lijke lijsten kunnen worden gegroepeerd tot samengestelde lijsten, zoals inhoudsopgaven. Lijsten en samengestelde lijsten kunnen op ieder niveau worden opgeroepen. Binnen een hoofdstuk krijgt men dus alleen de paragrafen, subparagrafen enz. te zien. Bij interactieve teksten zijn de lijsten automatisch doorgelust, zodat men snel door een tekst kan lopen (van inhoud naar subinhoud naar subsubinhoud enz.).

Omdat in dit document een structuur in termen van hoofdstukken (secties) niet erg voor de hand ligt, is gekozen voor het genereren van ruim vijftig afzonderlijke lijsten.

Via de tussenstap *publicaties* (een lijst) komt men bij de afzonderlijke lijsten met publicatienummers. Een kritische lezer zal zich afvragen of het tonen van nummers zin heeft. Welnu, veel van de gebruikers zijn gewend om met nummers om te gaan en vertalen nummers automatisch naar de betekenis.

Bij de publicatie vinden we links onder de categorie en rechts het trefwoord. Omdat er meerdere publicaties per categorie en trefwoord mogelijk zijn, wordt de mogelijkheid geboden naar verwante publicaties te springen. In een interactief document van deze omvang is het namelijk vrij zinloos om achter een ingang in een register enige tientallen paginanummers te tonen. Als we op het woord klikken keren we terug naar de relevante bladzijde in het register.

Overigens geldt voor registers een vergelijkbaar verhaal als voor lijsten. Er kan een onbeperkt aantal registers worden gedefinieerd, die op elk niveau als deelregister kunnen worden opgeroepen (per deel, hoofdstuk, enz. of per zelf te definiëren sectie).

Het eerste register meldt de categorieën en is dus een variant op de eerder besproken lijst.

Het tweede register beslaat tien bladzijden, wat te zien is aan de twee na onderste navigatiebalk. Het blokje dat bij de actuele bladzijde hoort is rood, de rest is groen. De balkjes krijgen een andere vorm als de afstand ertussen onder een kritische grens komt. De reden hiervoor ligt in het afnemend oplossend vermogen van het aanklikbare deel en de onnodig grote overhead in termen van bytes die dergelijke sprongen met zich meebrengen.

Er zijn een, twee of drie opties mogelijk. Als er slechts één verwijzing is, dan tonen we een blokje. Als er twee zijn, dan tonen we twee pijltjes, waarmee men naar het eerste of laatste trefwoord kan springen. Bij drie of meer verwijzingen gaan we met het middelste blokje naar het midden van de reeks bij het trefwoord behorende publicaties. Daar aangekomen kan men stapsgewijs de publicaties langs lopen, wat handiger is dan steeds weer terug te moeten keren naar het register.

TREFWOORDEN (1)		publicatie
AANHANGWAGEN	ARTIKEL	publicatie
AANSCHAFFING	ARTILLERIE	publicatie
AANSTELLING	AUTOBAND	publicatie
AANTEKENVEL	AUTOMATISERING	publicatie
AANVRAAG	AUTORISATIE	publicatie
AANVULLINGSPLTS	AVI	toelichting
AANWIZING	AVIB	toelichting
ACCU	BATALJON	helpinfo
ADMINISTRATIE	BATTERIJ	colofon
ADRES	BEDIENING	colofon
AFKORTINGEN	BEDRIJF	colofon
AFLEV.STAAT	BEDRIJFSSTOFFEN	colofon
AFVOEREN	BEDRIJFSUREN	colofon
AGGREGAAT	BEGELEIDING	colofon
AIP	BELADING	colofon
ANALIST	BELANGSTELLING	colofon
ANATOMIE	BEOORDELING	colofon
ANTI-TANK	BEREKENING	colofon
APPARATUUR	BERGING	colofon
APPELLIJST	BERICHT	colofon
ARBEID	BESCHERMING	colofon
ARBO	BESCHIKKING	colofon
ARMSEIN	BESLAGLEGGING	colofon

Bij omvangrijke documenten komt Acrobat in de problemen als we gebruik maken van de /dest operator onder pdfmark's. Omdat de tekst geschikt moet zijn om onder MSDOS te raadplegen, is echter gebruik gemaakt van het springen naar bladzijden. Omdat de MSDOS versie achterloopt in de ontwikkeling van pdfmark's, is springen naar andere documenten niet mogelijk.

We kunnen dus niet werken in termen van *hier ben ik en ga daar naar toe*, maar moeten het doen met *ga naar bladzijde*. Dit betekent dat alle relevante paginanummers tijdens de verwerking door T_EX beschikbaar moeten zijn. Met wat kunstgrepen lukt dat vrij aardig, zij het dat Huge T_EX onmisbaar is.

3 Het tweede document

Het tweede document is niet alleen veel omvangrijker (bijna 40.000 records) maar ook veel gecompliceerder. Het betreft hier namelijk in projecten gegroepede publicaties, waarbij binnen een groep onderscheid wordt gemaakt in hoofdpublikaties en 'verborgen' publicaties.

ZENDER, DOELINSTALLATIE, IR, M89E1 DRAGO N WAPENSYSTEEM		KL-nummer
kl-artikelnummer : KL0074682	projectnummer : 006097	project
NATO-stocknummer : 6920 - 000714482	kl-typenummer :	NATO-stock
1 DL 6920-00-071-4482 07-07-1994	ZENDER, DOELINSTALLATIE, IR, M89E1	publicatie
1 TH 9-0249/1 01-12-1980	DRAGON WAPENSYSTEEM M47 OPLEIDINGSAPPARATUUR GEDEELTE ZENDERDOELINSTALLATIE, IR, M89E1	toelichting
25 DL 000249	DRAGON WAPENSYSTEEM M47 OPLEIDINGSAPPARATUUR	helpinfo
1 DL 6920-17-051-4979	DRAGON WAPENSYSTEEM M47 OPLEIDINGSAPPARATUUR, TRAINER, LANCEEREFPECTEN, M54, MONITOR, INSTALLATIE, ZENDER, DOELSTELLING, IR, M89E1, OEFENTOESTEL, EXERCITIE	colofon

Omdat het niet realistisch is ruim 40.000 bladzijden te genereren, is besloten de gegevens te groeperen. Daarbij is onderscheid gemaakt in algemene informatie, hoofdpublikaties en verborgen publicaties. Het bestand is daartoe gesorteerd, waarna aan T_EX de zorg voor de nadere ordening en controles op consistentie is overgelaten. In tegenstelling tot het eerste bestand kan een groep publicaties zich hier

uitstrekken over meerdere bladzijden. Ondanks de compacte vorm komen we toch nog op bijna 8.500 pagina's uit.

PROJECT (1)					KL-nummer
000001	000060	000097	000137	000176	publicatie
000002	000061	000098	000138	000177	publicatie
000007	000062	000099	000141	000178	NATO-stock
000017	000063	000100	000143	000181	publicatie
000029	000064	000101	000144	000183	publicatie
000030	000066	000102	000146	000184	publicatie
000031	000068	000103	000147	000188	publicatie
000032	000069	000104	000149	000189	publicatie
000033	000070	000105	000150	000198	publicatie
000034	000071	000106	000152	000199	publicatie
000036	000077	000107	000154	000201	publicatie
000037	000079	000108	000156	000204	publicatie
000038	000082	000109	000157	000215	publicatie
000041	000083	000110	000158	000216	publicatie
000042	000085	000111	000159	000217	publicatie
000043	000088	000112	000160	000218	publicatie
000047	000090	000113	000161	000219	publicatie
000053	000091	000116	000162	000220	publicatie
000054	000092	000117	000163	000221	publicatie
000055	000093	000118	000172	000224	publicatie
000057	000094	000120	000173	000225	publicatie
000058	000095	000123	000174	000226	publicatie
000059	000096	000136	000175	000227	publicatie

In dit document zijn alleen registers gebruikt. Veel ruimte is daarbij bespaard door te werken in kolommen. Het aantal kolommen wordt binnen CONTEXT alleen begrensd door het maximaal aantal te alloceren boxen. Om te voorkomen dat de registers onoverzichtelijk lang worden, zijn verborgen publicaties niet opgenomen.

Bij documenten met een dergelijke omvang, komen we uit bij iets anders vormgegeven navigatie-instrumenten. Ook hier zien we een uit blokjes opgebouwde balk rechts-onder. In dit geval is de resolutie echter (automatisch) aangepast aan de beschikbare ruimte. Men kan steeds naar de eerste en laatste bladzijde van een subpaginabereik springen. Bovendien kan men naar de aangrenzende bladzijden gaan en in grote sprongen naar de overige. Al bladerend zien we in zo'n balk de wat hogere blokjes van links naar rechts schuiven.

Een van de registers heeft sub-ingangen. Dit wat cryptische register is voor kenners wel hanteerbaar, omdat de gebruikers bij de nummers een soort document voor de geest kunnen halen.

NATO-STOCK-NUMMER (67)			KL-nummer
6545 170244125	6545 170248568	6545 170447779	project
6545 170244144	6545 170248569	6545 170465059	project
6545 170244152	6545 170248570	6545 170465098	NATO-stock
6545 170244205	6545 170248571	6545 170466907	publicatie
6545 170244206	6545 170248572	6545 170468137	publicatie
6545 170244526	6545 170248578	6545 170468159	publicatie
6545 170244962	6545 170258076	6545 170468251	publicatie
6545 170244994	6545 170258077	6545 170468266	publicatie
6545 170244995	6545 170259216	6545 170468851	publicatie
6545 170245094	6545 170259477	6545 170468922	publicatie
6545 170245134	6545 170260359	6545 170469476	publicatie
6545 170245271	6545 170297027	6545 170469479	publicatie
6545 170245428	6545 170321692	6545 170469725	publicatie
6545 170245844	6545 170321875	6545 170469737	publicatie
6545 170246056	6545 170321883	6545 170480978	publicatie
6545 170246177	6545 170322808	6545 170481344	publicatie
6545 170246178	6545 170322813	6545 170483244	publicatie
6545 170246180	6545 170322814	6545 170484690	publicatie
6545 170246186	6545 170336324	6545 170524801	publicatie
6545 170246187	6545 170344831	6545 170537150	publicatie
6545 170246586	6545 170344881	6545 170537151	publicatie
6545 170246588	6545 170422230	6545 170537152	publicatie
6545 170246595	6545 170437668	6545 170537153	publicatie

4 Het bestelformulier

Tot slot is er nog het bestelformulier. De getoonde variant is een vrijwel een-op-een vertaling van het bestaande document en wordt dan ook in A4 formaat getoond. Klik

men op een veld, dan wordt de bijbehorende helpinformatie getoond. Van daaruit kan men weer terugspringen naar het formulier. Helaas gaat binnen Acrobat Reader de cursor niet op de bestemming staan, zodat de gebruiker (met zijn ogen) weer het veld moet zoeken.

PUBLICATIE-NUMMER (3)			KL-nummer
1/2DL 5810-17-030-9001	5895 - 010166961	1/2DL 6230-17-107-4708	project
5810 - 170309001	1/2DL 5985-17-036-6397	6230 - 171074708	NATO-stock
1/2DL 5810-17-043-0301	5985 - 170366397	1/2DL 6515-17-046-8293	publicatie
5810 - 170430301	1/2DL 5985-17-036-9860	6515 - 170468293	toelichting
1/2DL 5810-17-046-1980	5985 - 170369860	1/2DL 6530-17-055-8748	helpinfo
5810 - 170461980	1/2DL 6130-17-047-2070	6530 - 170558748	colofon
1/2DL 5815-17-035-7787	6130 - 170472070	1/2DL 6605-17-047-7448	
5815 - 170357787	1/2DL 6130-17-103-2847	6605 - 170477448	
1/2DL 5815-17-035-7788	6130 - 171032847	1/2DL 6625-01-063-6628	
5815 - 170357788	1/2DL 6130-17-103-3175	6625 - 010550802	
1/2DL 5825-17-106-0045	6130 - 171033175	6625 - 010636628	
5825 - 171060045	1/2DL 6130-17-106-6194	6625 - 011134262	
1/2DL 5855-17-035-2846	6130 - 171066194	6625 - 170524551	
5855 - 170352846	1/2DL 6145-00-226-8812	6625 - 170525653	
1/2DL 5855-17-049-5863	6145 - 002268812	6625 - 170540600	
1450 - 010274616	1/2DL 6145-17-036-9410	1/2DL 6625-17-038-8807	
2350 - 010274623	6145 - 170369410	6625 - 170388807	
2350 - 010274624	1/2DL 6230-17-050-7591	1/2DL 6625-17-046-8863	
2350 - 010290637	6760 - 170507591	6625 - 170468863	
2350 - 010310311	6760 - 170540838	1/2DL 6625-17-051-6134	
5855 - 170495863	6780 - 170529689	6625 - 170516134	
1/2DL 5895-01-016-6961	1/2DL 6230-17-056-1980	1/2DL 6625-17-051-6656	
5820 - 170352178	6230 - 170561980	6625 - 170516656	

5 Enkele opmerkingen

Tijdens het ontwerpen van dergelijke documenten ontkomt men niet aan het op punten uitbreiden en optimaliseren van de functionaliteit van de gebruikte systemen.

We gaven reeds aan dat 'hugé' T_EX noodzakelijk is. Hetzelfde geldt voor het hulpprogramma dat de registers sorteert. Nog niet vermeld is dat CON_TE_XT slechts één hulpfile gebruikt. Dat wil zeggen dat alle lijsten, registers, synoniemlijsten, verwijzingen en andere noodzakelijke informatie, zoals subpaginaclusters in een file worden geplaatst. Deze file wordt door T_EXUTIL omgezet in een tijdens de volgende verwerkingsslag in te lezen hulpfile, waarin de in aanmerking komende lijsten en registers zijn gesorteerd. CON_TE_XT haalt op gezette tijden de nodige informatie uit deze file.

Aangezien de behoefte de capaciteit van T_EXUTIL ruim te boven ging is de capaciteit aanzienlijk vergroot. De bijna veertigduizend te verwerken ingangen in de registers van het tweede document liggen momenteel ruim onder de eenvoudige te verhogen limiet van 250.000 in T_EXUTIL.

Het karakter van CON_TE_XT brengt met zich mee dat relatief veel rekentijd wordt opgeofferd aan de gebruikersinterface. Desondanks blijft onze voorkeur uitgaan naar gebruik van standaard CON_TE_XT mogelijkheden boven speciaal ontwikkelde macro's. Wel is zo'n project een goede gelegenheid de kritische onderdelen van een macropakket te optimaliseren. Zo leverde bijvoorbeeld de optimalisatie van het mechanisme dat de actuele kleuren over de paginagrenzen tilt, een verdubbeling van de verwerkingssnelheid op.

Het zoekmechanisme van Acrobat Reader is bij dergelijke files nauwelijks te gebruiken. Ten eerste is de omvang van de files te groot, daarnaast is het lastig zoeken in de bonte mix van woorden, afkortingen en getallen. Om eenzelfde reden kan ook geen gebruik worden gemaakt van Acrobat

Catalog. (Wat onder MSDOS overigens niet beschikbaar is.) Een lijst met niet in een index op te nemen woorden is nauwelijks op te stellen. Bovendien is de omvang van deze lijst beperkt. Een ander nadeel is dat de gegenereerde bestanden bijna net zoveel ruimte in beslag nemen als de tekst zelf. Het blijft dus T_EX- werk.

AANVRAGEN / INLEVEREN PUBLICATIES

gebruikerscode datum

naam aanvrager aanvraagnummer

telefoonnummer

Lees eerst de invulinstructie!

aanvragen inleveren

1	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

2	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

3	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

4	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

5	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

6	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

7	elconribwstn	publicatienummer	wijziging	soort	nieuw	huidig	aantal
							autorisatie
	reden						

handtekening

Om de gegevens te sorteren is een programma geschreven waarmee zeer omvangrijke bestanden kunnen worden gesorteerd op meerdere te specificeren velden. Gelukkig zijn dergelijke programma's acceptabel snel te programmeren.

Tegenover het voordeel van platform-onafhankelijkheid en typografische kwaliteit staat het nadeel van omvangrijke files. De verwerkingstijd van dergelijke omvangrijke bestanden is niet onaanzienlijk, ook bij gebruik van snelle systemen. Bovendien worden grote tussenproducten gegenereerd, zodat ook veel schijfruimte nodig is. Vandaar dat de meeste verwerkingslagen 's nachts in batch hebben plaatsgevonden.

bestand	DVI	PS	PDF	blz
debkl-ep	19.578	22.131	15.351	3.046
debkl-vs	53.688	62.611	42.258	8.477
debkl-ef	30	55	29	4
gezipped	3.458	3.505	20.602	

In de bijgaande tabel is in KiloBytes de omvang van de verschillende files vermeld. De omvang van de DVI-files is mede bepaald door de aanwezigheid van de vele specials voor kleur en hyperlinks. Daarbij kan worden opgemerkt dat zowel Acrobat als DVIWINDO wordt onder-

steund. In de PostScript-file zijn alleen pdfmark's opgenomen. De PostScript-file wordt aan Acrobat Distiller aangeboden zonder font-outlines.

De PDF-file is, wat we ook mogen verwachten, kleiner dan de PostScript-file. Omdat we 'partial font embedding' toepassen, zijn hierin wel de fonts (deels) opgenomen. De omvang van PDF-files wordt echter hoofdzakelijk bepaald door het aantal sprongen. Op dit moment neemt Acrobat Distiller pdfmark's bijna verbatim in de PDF-file over, terwijl andere informatie wordt gecomprimeerd. Mijns inziens is een besparing van 50 bytes per sprong zonder problemen te realiseren. Wat dit oplevert bij vele tienduizenden sprongen laat zich eenvoudig uitrekenen.

Een aardige bijkomstigheid is dat de PDF-file zich veel minder efficiënt laat comprimeren dan de DVI- en PostScript-files. Waarschijnlijk heeft dit te maken met de wijze waarop binnen het PDF-formaat de gegevens worden gestructureerd.

Wellicht kan de toegankelijkheid worden vergroot door gebruik te maken van gerichte plug-ins, bijvoorbeeld om registers te doorlopen. Meer dan een toevoeging kunnen deze echter niet zijn, omdat we afhankelijk zijn van de beschikbaarheid ervan.

6 Tot slot

Kijkend naar de aard en omvang van het document en het aantal verwijzingen dat daarvan het gevolg is, kunnen we onszelf de vraag stellen wanneer we de grens van het redelijke overschrijden. Met name het tweede hierboven beschreven document zit dicht tegen deze grens aan. De doorslaggevende redenen om gebruik te maken van Acrobat liggen vooral in de beschikbaarheid op meerdere platforms en het onderliggende bestands-formaat. T_EX voegt daar nog een meer dan gemiddelde typografische kwaliteit aan toe.

Paradigms: It's all in the game

Kees van der Laan

1 BLUE's Design VI

Hi folks. In 1989 Greene contributed 'Playing in T_EX's mind' to TUG 89. Having fun with T_EX has been on my mind all the time. Of late, I thought of tic-tac-toe as an exercise in programming dialogues in T_EX, in a simple and elegant way, with hopefully some paradigms emerging, suited for educational purposes. The play is simple but nevertheless entails essential aspects in dialoging. Below I first strip the play to the essentials and from there build the user-friendly but otherwise modest code. At the end the T_EXing peculiarities are summarized.

2 The play

Tic-tac-toe is played by two persons on a board with 3×3 fields. Each player marks a field in turn and the one who has first three marks in a row has won.

+	o	+
	o	+
o		+

3 Prototyping

The dialogue with T_EX goes through the log file, also called transcript.

The board is represented by the defs \1, ... \9 row-wise. Each player owns a mark called \markplayer and \markopponent. A user is prompted to input a value—1, 2, ... or 9—which indicates the field where to put his mark. The program keeps track of the kind of mark, via a toggle. To end the game the user can input 0.

The above resulted in the following bare-to-the-bones implementation of tic-tac-toe.

```
\def\showboard{\immediate\write0{\1\2\3}
\immediate\write0{\4\5\6}
\immediate\write0{\7\8\9}}
\def\initialize{\def\1{-}\def\2{-}\def\3{-}
\def\4{-}\def\5{-}\def\6{-}
\def\7{-}\def\8{-}\def\9{-}}
\def\play{\initialize \loop\showboard
\ifx\mark\markplayer
\let\mark\markopponent\else
\let\mark\markplayer\fi
\immediate\write0{Supply index for \mark:}
\read0to\index \expandafter
\xdef\curname\index\endcurname{\mark}
\ifnum\index>0 \repeat}%end \play
\def\markplayer{+}\def\markopponent{o}
\endlinechar-1 %TB20.18
\play \bye
```

¹Not that difficult, actually, because the set of allowed indices is maintained.

Remarks. Because of the toggling of the \mark an \xdef was needed.

One could code that the player and opponent are both taking care of during each traversal of the loop. However, apart from that it makes the code longer it is also clumsy, and if careless one has to account for multiple exits of the loop. I assumed states in which the loop is traversed. The next steps are not for the faint of heart.

4 Real-life version

Given the above prototype implementation we can refine, add more bells-and-whistles. Basically these add-ons are in two directions

- improve the user interface
- let the program do more, such as deciding who has won.

4.1 Improving the user interface

Explain the conventions adopted, especially what the indices stand for.

Always nice is to allow for personalization, i.e. the system prompt asks for YOU, with your name. This entails that the players must be asked to identify themselves and that the toggling must be extended.

Another feature is that the program prompts the remaining indices to choose from.

And what about robustness? I decided not to implement robustness with respect to lowercase or uppercase y or n, for example. I also refrained from checking whether the supplied index is allowed.¹

4.2 Let the program do more

The most important aspect is to add intelligence to the program to check for a winner.

The play restarts automatically.

4.3 The code

The board is represented by the defs \1, \2, ... \9, the 3×3 board row-wise.

The structure of the program is similar to the prototype, with \play elaborated and the check \checkforgameend added.

```
\immediate\write0{Tic-tac-toe
Aug 1995, cgl@rc.service.rug.nl}
\let\ea\expandafter
```

```

\newcount\k\newcount\value\newcount\checksum
\newif\ifsol
\newtoks\set %Index set
\def\del#1{\def\lop##1#1##2\pol{\set{##1##2}}
  \ea\lop\the\set\pol}
%
\def\showboard{\immediate\write0{}}
\immediate\write0{\1\2\3}
\immediate\write0{\4\5\6}
\immediate\write0{\7\8\9}}
%Initialization
\def\initialize{\set{123456789}\solfalse
  \immediate\write0{New names of players?
    (default \player\space and \opponent)}
  \read0to\yorn
  \if y\yorn
    \immediate\write0{Name player}
    \read0to\player
    \immediate\write0{Name opponent}
    \read0to\opponent
  \fi\k0
  \loop\advance\k1
    \ea\def\csname\the\k\endcsname{-}
    \ifnum\k<10
  \repeat
  \immediate\write0{Empty board}
}%end initialization
%Test for solution
\def\sol#1#2#3{{\advance\count#1\count#2
  \advance\count#1\count#3
  \ifnum\count#1=\checksum \global\soltrue\fi}}
\def\checkforgameend{%
\sol123\ifsol\message{\who\space won}\k0 \else
\sol456\ifsol\message{\who\space won}\k0 \else
\sol789\ifsol\message{\who\space won}\k0 \else
\sol147\ifsol\message{\who\space won}\k0 \else
\sol258\ifsol\message{\who\space won}\k0 \else
\sol369\ifsol\message{\who\space won}\k0 \else
\sol159\ifsol\message{\who\space won}\k0 \else
\sol357\ifsol\message{\who\space won}\k0 \else
\fi\fi\fi\fi\fi\fi\fi}
%Play
\def\play{\initialize\beginngroup
\loop\showboard
  \ifx\who\player\value-1 \checksum-3
    \let\who\opponent
    \let\mark\markopponent
  \else\value1 \checksum3
    \let\who\player
    \let\mark\markplayer
  \fi
  \immediate\write0{\who, supply index
    for \mark:}
  \immediate\write0{Choose from: \the\set.
    (0 terminates)}
  \read0t\ea o\csname\who\endcsname
  \k\csname\who\endcsname
  \ea\xdef\csname\the\k\endcsname{\mark}
  \count\k\value
  \checkforgameend
\ifnum\k>0 \ea\del\ea{\the\k}
\repeat\endgroup
\immediate\write0{Play another game?}
\read0to\newplayyorn
\if y\newplayyorn\ea\play\fi}%end Play
%Defaults
\def\player{Kees} \def\opponent{Ina}
\def\markplayer{+} \def\markopponent{o}
%
\immediate\write0{}
\immediate\write0{Board numbering}
\immediate\write0{123}

```

```

\immediate\write0{456}
\immediate\write0{789}
\play \bye

```

Remark. In order to get the personalized prompts \Kees, or \Ina the following coding was needed.

```
\read0t\ea o\csname\who\endcsname
```

The few lines that follow look unnecessary complex but are entailed by the above.

5 What more?

It is intriguing to ponder about adding even more intelligence. For example to let the game prompt for obligatory moves, or to let the program terminate when draw is inevitable, that is when there is no possible solution left. In order to achieve this I chose to

- maintain the set of possible solutions, instead of checking all possible solutions²
- update the set of solutions after each move, and look for a solution or a draw
- look for obligatory moves.

With respect to robustness the input can be checked for whether the index is allowed, *casu quo* a y(es) or n(o).

```

\immediate\write0{Tic-tac-toe
  Aug 1995, cgl@rc.service.rug.nl}
\let\ea\expandafter \let\nx\noexpand
\newcount\k\newcount\kk
\newcount\value\newcount\checksum
\newcount\feasible\newcount\prompt
%Solution lines
\newcount\hi\newcount\hii\newcount\hiii
\newcount\vi\newcount\vii\newcount\viii
\newcount\di\newcount\dii
\newif\ifsol
\newif\ifnotfound
%Index set and deletion from index set
\newtoks\set %Index set
\def\del#1{\def\lop##1#1##2\pol{\set{##1##2}}
  \ea\lop\the\set\pol}
\def\showboard{\immediate\write0{}}
\immediate\write0{\1\2\3}
\immediate\write0{\4\5\6}
\immediate\write0{\7\8\9}}
%Initialization
\def\initialize{\set{123456789}\solfalse
  \hi0 \hii0 \hiii0 \vi0 \vii0 \viii0
  \di0 \dii0 \feasible8 \prompt0
%Cell no associated with solution subsets
\ea\def\csname solset1\endcsname{%
  \ls\hi\ls\vi\ls\di}
\ea\def\csname solset2\endcsname{%
  \ls\hi\ls\vii}
\ea\def\csname solset3\endcsname{%
  \ls\hi\ls\viii\ls\dii}
\ea\def\csname solset4\endcsname{%
  \ls\hii\ls\vi}
\ea\def\csname solset5\endcsname{%
  \ls\hii\ls\vii\ls\di\ls\dii}
\ea\def\csname solset6\endcsname{%
  \ls\hii\ls\viii}
\ea\def\csname solset7\endcsname{%
  \ls\hiii\ls\vi\ls\dii}
\ea\def\csname solset8\endcsname{%
  \ls\hiii\ls\vii}
\ea\def\csname solset9\endcsname{%
  \ls\hiii\ls\viii\ls\di}
\immediate\write0{New names of players?

```

²This entails that non-feasible candidates are eliminated from the set of candidate solutions.

```

    (default \player\space and \opponent))
\read0to\yorn
\if y\yorn
  \immediate\write0{Name player}
  \read0to\player
  \immediate\write0{Name opponent}
  \read0to\opponent
\fi\k0
\loop\advance\k1
  \ea\def\csname\the\k\endcsname{-}
  \ifnum\k<10
  \repeat
  \immediate\write0{Empty board}
}%end initialization
%Test for solution
\def\ls#1{#1 is a solution counter
  \ifnum#1=0 \advance#1\value
  \else\ifnum\sign#1=\value \advance#1\value
    \else\advance\feasible-1
    \delete#1 %from solution sets
    \ifnum\feasible=0 \showboard
    \message{***Draw***}\k0
    \fi
  \fi
\fi
%Check for solution
\ifnum#1=\checksum \showboard
  \message{***\who*** won}\k0
\fi}
\def\sign#1{\ifnum#1>0 1\else -1\fi}
\def\solsetk{\csname solset\the\k\endcsname}
\def\delete#1{#1 is a solution counter to be
  %deleted from all solution sets
}\kk0
\def\ls##1{\ifx#1##1\else\nx\ls\nx##1\fi}
\loop\advance\kk1
\ifnum\kk<10
\ea\xdef\csname solset\the\kk\endcsname
  {\csname solset\the\kk\endcsname}
\repeat}}%end \delete
\def\fifo#1{\ifx#1\ofif\ofif\fi
  \process#1\fifo}
\def\ofif#1\fifo{\fi}
\def\lstry#1{#1 is a counter denoting
  %a solution
  \ifnum#1=0
  \else\ifnum\sign#1=\value
    \advance#1\value
    %Check for solution
    \ifnum#1=\checksum \global\prompt\k
    \fi\fi
  \fi}
\def\process#1{\ifnum\prompt=0
  {\k=#1 \let\ls\lstry \solsetk}\fi}
\def\readprocess#1{\if#1\csname\who\endcsname
  \notfoundfalse\fi}
\def\readindex{\let\process\readprocess
  \loop\read0t\ea o\csname\who\endcsname
  %\who value in \set?
  \notfoundtrue\ea\fifo\the\set0\ofif
  \ifnotfound
  \immediate\write0{Please supply index
    from \the \set}
  \repeat
  \global\k\csname\who\endcsname}}
%Play
\def\play{\initialize\beginngroup
\loop\showboard%\show\ the\solset
  \ifx\who\player\value-1 \checksum-3
  \let\who\opponent
  \let\mark\markopponent
  \else\value1 \checksum3
  \let\who\player
  \let\mark\markplayer

```

```

  \fi
  %Is there a winning move?
  %(Prefails obligatory move)
  {\prompt0 \ea\fifo\the\set\ofif
  \ifnum\prompt>0 \global\prompt0 \fi}
  %The idea is that the player must see it
  %him/herself. Of course the program knows
  %about the winning move.
  %Obligatory move? (criterion 0<\prompt (<10)
  \ifnum\prompt>0 \immediate\write0{Sorry \who,
    obligatory move \the\prompt}\k\prompt
  \else
  \immediate\write0{\who, supply index
    for \mark:}
  \immediate\write0{Choose from: \the\set.
    (0 terminates)}
  \readindex
  \fi
  %put mark on board
  \ea\xdef\csname\the\k\endcsname{\mark}
  %update solutions associated with \k
  \solsetk
  %k=0 is stopping criterion
  \ifnum\k>0 \ea\del\ea{\the\k}%
  %Look ahead for obligatory move:
  % Can \who gain in next turn?
  \prompt0 \ea\fifo\the\set\ofif
  \repeat\endgroup
  \immediate\write0{Play another game?}
  \read0to\newplayyorn
  \if y\newplayyorn\ea\play\fi}%end Play
%Defaults
\def\player{Kees} \def\opponent{Ina}
\def\markplayer{+} \def\markopponent{o}
%And off we go
\immediate\write0{
\immediate\write0{Board numbering}
\immediate\write0{123}
\immediate\write0{456}
\immediate\write0{789}
\immediate\write0{
\play \bye

```

6 Is this all?

Especially the code in the last section has become complex. Should we make it more complex? For the moment I stopped. However, it is tempting to increase the order to 4, and to extend the play to 3 dimensions. My case rest.

7 Paradigms

The following functionalities have been encountered and a way how to code these in \TeX have been worked out.

- loop traversal in various states
- global Boolean, $\global\soltrue$
- maintaining an index set, deleting an element from a set
- test for an empty set
- high-level parameterization, for example a personalized prompt within a state-dependent loop
- \read with a *reference* to a prompt through $\read0 t\ea o\csname\who\endcsname$
- check on what is read is allowed.

Starting from the bare-to-the-bones prototype it is quite something to arrive at a solution, which carries some intelligence and robustness .

Have fun, and all the best.

Paradigms: The winds and halfwinds—Details matter

Kees van der Laan

1 BLUE's Design V

Hi folks. How to draw lines at 45° in T_EX is exercised. This simple-looking and innocent problem touches on fundamental issues, such as should we adhere to the Cartesian picture environment approach,¹ or do we need something else now and then?

This work emerged from studying Gurari's approach to graphics incited by a suggestion in Jos Winnink's review, MAPS 94.2.

The discussion below explains what has been used in BLUE's format system in the turtle graphics² macros, especially in the coding of \N, \E, \S, \W, \NE, \SE, \SW, \NW, next to \ESE, and \WSW. Examples are included which show what can be attained by these basic functionalities. It is also shown how the obtained pictures can be stored in the picture database `pic.dat` for reuse with different scaling or positioning.

In the Appendix some tree diversions have been given.

2 What is the problem?

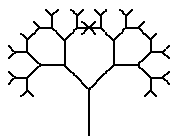
T_EX's \hrule and \vrule primitives are a gem and very powerful. It would be nice to have similar primitives for any direction. In absence of these we can use line pieces provided in fonts. However, the latter approach suffers from the following drawbacks

- for a few discrete directions only
- line lengths are discrete too
- line thickness is inflexible.

3 Why?

The need for general and flexible line elements arose when I faced the problem how to draw classical fractals.³ The very least is the possibility to draw lines at 45°—the halfwinds.

Example (*Pythagorean tree*)



How to do this in T_EX⁴?

¹The L^AT_EX picture environment for example.

²Knuth already used the 'Turtle' idea in his dragon figures. For those interested in turtle graphics consult for example S. Papert: Mindstorms.

³Inspired by Gurari's work on T_EX and graphics.

⁴For the coding of \pythtree see later. Or Macro writing—with confidence.

⁵J V Romanovsky has provided a concise POSTSCRIPT program.

⁶The halfwinds can be composed from the winds in this case.

Through the picture environment? Too clumsy, and cumbersome when changing the order, for example.

Through POSTSCRIPT? This is possible, especially it provides for lines at all orientations.⁵

Through METAFONT? Definitely, see my METAFONT anthology.

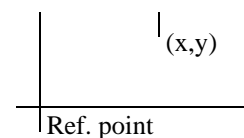
However, why not go for how far we can get just by T_EX?

And what about the relevancy? I'm very pleased by the spin-off how to typeset binary trees or charts, even rotated without the use of POSTSCRIPT. See the Appendix or the Publishing with T_EX user's guide.

What we need is not the Cartesian picture environment approach, but the good old penplotter T_EXniques, better known in the pedagogical world as 'Turtle graphics.'

4 Turtle graphics

The basic idea is that the position of the turtle is maintained in the dimen variables \x and \y, with the reference point left invariant. The accompanying figure shows the effect of \N1: draw the line (\x, \y)–(\x, \y+1), i.e. the turtle moves up. After completion \y has been increased by \unitlength.

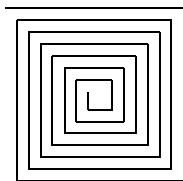


The movements—our first steps in the turtle graphics world—can be achieved by the control sequences

- \N, \E, \S, \W mean draw north, east, south and west, similar for \NE, \SE, \SW, and \NW
- \whiteN, \whiteE, \whiteS, \whiteW mean *white*-draw north, east, south and west, i.e., the turtle just moves.⁶

To get the flavour another classical picture, which shows that we don't have to worry about coordinates.

Example (*Spiral*)



The spiral is marked up essentially as follows ⁷

```
\unitlength.5ex \k30
\loop\E{\the\k}\advance\k-1
  \S{\the\k}\advance\k-1
  \W{\the\k}\advance\k-1
  \N{\the\k}\advance\k-1
\ifnum\k>4 \repeat
```

5 Basics

Essential for the understanding is familiarity with T_EX's *boxes of size zero*, to know the effect of `\kern`-s and `\h/vss`-s inside, and of combinations of these boxes. Before reading on peruse the following.

5.1 Kerning indispensable

Kerns and stretchorshrinks within boxes of size zero are at the heart of graphics with T_EX alone.

Example (*With and without boxes of size zero*)

```
\newdimen\x \x=4ex \newdimen\y \y=2ex
.\hbox to 0pt{\kern\x a\hss}.
\kern10ex and \kern10ex
.\kern\x a.
```

and

```
.\hbox to 0pt{\kern\x\vbox to 0pt
  {\vss\hbox{a}\kern\y}\hss}.
\kern10ex and \kern10ex
.\kern\x\raise\y\hbox{a}.
```

with result

```
.. a and . a.
```

and

```
.. a and . a.
```

By this mechanism we can move to any point on the page and put there what we wish. The essential issue of the box of zero width is that the *reference point is left invariant*. It is the same before and after.

5.2 Putting it together

In vertical mode the `\hbox`-s are aligned on the reference point, and when the heights and depths are zero the `\hbox`-s overprint, and the order of specification is immaterial.

In (restricted) horizontal mode `\hbox`-s of width zero overprint and can be given in any order, The T_EXbook 389.

In math mode the zero size boxes overprint. In display math the invariant reference point is centered horizontally.

Coding in T_EX simple? Forget it. So unusual, but ...consequent, utmost consequent.

5.3 Coding flashbacks

The codes which follow are the fifth or so version. The zeroth version was there to please me, it made me excited. Recursive programming worked as I expected. Then I had a version which served to try out as many examples as I could think of. During the process of collecting examples I strove after consistency of the codes. Why did I use integer arithmetic anyway? I had no good reason and I decided to use dimensions and do the arithmetic through the dimensions, and widened en-passant the allowed arguments of the winds and halfwinds to expand to 'factor'-s.

And what about the context?

It seemed necessary to abstract from the multitude of boxes and to come up with an enveloping `\hbox` of width zero.

Finally, details were taken care of, such as how do the macros behave in the various contexts. Do the winds and halfwinds behave the same? It turned out that they did not. So I had to go back from where I started. I considered the early versions as prototypes and started anew guided by specs, with simplicity and robustness as yardsticks, stating precisely how the elements should behave. By all means no sinecure, but IMHO with all respect, a must in software engineering.⁸ Indeed, a superficially, simple problem.

Is that all? Essentially yes, but ...details matter.

6 The winds and halfwinds

The intuitive idea is to compose lines out of elements, from what I call atoms. I used squares, *casu quo* rectangles, as atoms, and stacked these as follows.⁹

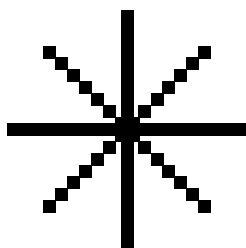
Essential is how lines leave a mathematical point. The accompanying model picture is obtained through the following macros.

```
\let\0\N \let\1\NE \let\2\E \let\3\SE
\let\4\S \let\5\SW \let\6\W \let\7\NW
\linethicknesslex
\setbox\hlfwndelm=\hbox{\vrule
  width\the\linethickness
  height\the\linethickness depth0pt}
\unitlength10ex
\def\draw{\csname\the\dir\endcsname1}
$$\loop\ifnum\dir<8{\draw}\advance\dir1\repeat$$
```

⁷For the placement within context details consult `\spiralspic` in `pic.dat`.

⁸In fact I had this article nearly finished when I realized that the vertical rules induce a `\vbox` of the size of the rules, while other directions when boxed yielded a box of size zero. Not nice and confusing, although for the article and in my hands it was ok. Nevertheless, I decided to go for a uniform approach. When boxed the dimensions of the box should be zero.

⁹In order to make it visible `\linethickness` has been set to `lex`. Experiments with bullets and L^AT_EX's line fonts did not yield the pleasing exact results.



Remarks. `\N` etc. draw in the implied directions with length as prescribed by the argument times `\unitlength`.

7 Pondering aloud

Can we attain compatibility with \TeX 's rule primitives? I don't think so, alas. On the other hand we have to reconsider the leaving of the rules from a point.

7.1 Thickness

What is meant by thickness if we stack instead of tile? What is the perceived blackness?

I used that stacking with square elements of size `\linethickness*\linethickness`—like in the example figure—yields the same blackness as a rule of thickness `\linethickness`.¹⁰

7.2 Size

Usually the size along the x-axis must be provided. I prefer to have the real size specified independently from the orientation of the line. However, the resulting size is not necessarily `#1*\unitlength`.¹¹ In general the result differs at most by half the atom size, because it is composed of a multiple of the atomic element. We have to correct by $\sqrt{2}$ to compensate for the direction, when we pace along one of the axes. In the example the required length is `10ex`, with as result the stacking of 7 atoms of size `1ex`.

For large lines one could think of combining \LaTeX 's `line10` line elements with the smaller elements. I refrained from that for two reasons. Firstly, the \LaTeX 's NW line element—the `\char'145`—did not fit exactly in the box, and secondly, the inflexibility of the thickness of the font elements.

8 Design specs

With awareness of the above I specified the following for the microscopic level—the wind and halfwind commands proper—and for the macroscopic level—the placement within context.

8.1 Microscopics

The functionality is to draw a line of the specified length in the direction as implied by the control sequence name. The general specifications read as follows.

- as argument a 'factor' is expected, to yield the required length `#1*\unitlength`¹²
- parameterization is by `\linethickness`
- after drawing, the position of the turtle is at the end of each line, the reference point has been left invariant, and all the boxes have zero width, height and depth.

Extra for the four halfwinds the following.

- parameterization is by `\hlfwndel` and `\linethickness` in there
- draw a line of \approx the specified size
- the atoms are stacked diagonally, at the corners.

8.2 Macroscopics

The placement within context is the concern of the user. Because of the zero dimensions of the boxes it is a nuisance to skip or kern when using a picture to create the open space, the niche for the picture. Moreover, when the picture does not take dimensions we are in trouble at page breaks. Therefore assistance is badly needed. The picture environment idea combined with databases come to rescue. The use of prefab pictures has been simplified in this way, while there is flexibility via `\thispicture` to override the defaults.

Pictures can be stored in BLUE's format `pic.dat` database. Within each database element the default placement within context can be provided for. Through the use of `\everypicture` and/or `\thispicture` the defaults can be overridden. This approach complies with the general principles adopted in BLUE's format system.

9 Coding the winds and halfwinds

In the fourth version, where emphasis was put on that all boxes have zero dimensions, I also decided to separate to get at the (x, y) position from what is put there. This is much in the spirit of the second `\point` macro of The \TeX book 389, and adheres to the *separation of concerns* adage.¹³

The dimension variables `\x` and `\y` have after completion the values of the coordinates of the end of the line.

```
\newbox\hlfwndel
\newdimen\linethickness \linethickness1ex
\newdimen\auxdim %linesize
%
\def\xy#1{%Function: place #1 at x, y
  \vbox to0pt{\kern-\y
    \hbox to0pt{\kern\x#1\hss}\vss}}
%
\def\xytxt#1{%Function: place text #1 at x, y
  \xy{\vbox to0pt{\vss
    \hbox to0pt{\strut#1\hss}\kern0pt}}
%
\def\N#1{\xy{\kern-.5\linethickness
  \vbox to0pt{\vss
    \hrule height#1\unitlength
    width\linethickness}}%
  \advance\y#1\unitlength}
```

¹⁰I also supplied macros—tiled variants—for `\NW` and `\NE`.

¹¹To put it in another way: the required length must be a multiple of the atom size.

¹²The idea is that not only integer values can be specified but also decimal fractions.

¹³IMHO it makes the code more trustworthy and circumvents pitfalls. Especially, the mixing up of the kerns needed to get at (x, y) with the kerns to position what has to be put at (x, y) .

```

%
\def\S#1{\advance\y-#1\unitlength{\N{#1}}}
%
\def\E#1{\xy{\vbox to0pt{\vss
  \hrule width#1\unitlength
  height\linethickness
  depth0pt\vss
}}\advance\x#1\unitlength}
%
\def\W#1{\advance\x-#1\unitlength{\E{#1}}}
%
\def\NE#1{\auxdim#1\unitlength\correction
\loop\advance\auxdim-\wd\hlfwndelm
\ifdim\auxdim>-.5\wd\hlfwndelm
  \xy{\vbox to0pt{\vss\copy\hlfwndelm}}%
  \advance\x\wd\hlfwndelm
  \advance\y\ht\hlfwndelm
\repeat}
%
\def\NW#1{\auxdim#1\unitlength\correction
\loop\advance\auxdim-\wd\hlfwndelm
\ifdim\auxdim>-.5\wd\hlfwndelm
  \advance\x-\wd\hlfwndelm
  \xy{\vbox to0pt{\vss\copy\hlfwndelm}}%
  \advance\y\ht\hlfwndelm
\repeat}
%
\def\SW#1{\auxdim#1\unitlength\correction
\loop\advance\auxdim-\wd\hlfwndelm
\ifdim\auxdim>-.5\wd\hlfwndelm
  \advance\x-\wd\hlfwndelm
  \advance\y-\ht\hlfwndelm
  \xy{\vbox to0pt{\vss\copy\hlfwndelm}}%
\repeat}
%
\def\SE#1{\auxdim#1\unitlength\correction
\loop\advance\auxdim-\wd\hlfwndelm
\ifdim\auxdim>-.5\wd\hlfwndelm
  \advance\y-\ht\hlfwndelm
  \xy{\vbox to0pt{\vss\copy\hlfwndelm}}%
  \advance\x\wd\hlfwndelm
\repeat}
%For \NE and \NW the titled variants read
\def\NE#1{\auxdim#1\unitlength\correction
\loop\advance\auxdim-.5\wd\hlfwndelm
\ifdim\auxdim>-.25\wd\hlfwndelm
  \xy{\vbox to0pt{\vss\copy\hlfwndelm}}%
  \advance\x.5\wd\hlfwndelm
  \advance\y.5\ht\hlfwndelm
\repeat}
%
\def\NW#1{\auxdim#1\unitlength\correction
\loop\advance\auxdim-.5\wd\hlfwndelm
\ifdim\auxdim>-.25\wd\hlfwndelm
  \xy{\vbox to0pt{\vss
    \llap{\copy\hlfwndelm}}}%
  \advance\x-.5\wd\hlfwndelm
  \advance\y.5\ht\hlfwndelm
\repeat}

```

10 Coding a database element

As promised in the Macroscopics section the visibility of the picture—that is the size—has to be added when the picture is stored in the database for (re)use. In order to do so I added two extra layers. The structure of a `pic.dat` database element, and a picture environment on top of for example `gkp`'s environment to allow user flexibility.

¹⁴Note that there is flexibility here. `\bintreepic` consists of an invoke of `\bintree` preceded scaling and positioning parameters, assigned with default values. The flexibility comes down to the possibility to override the defaults via the use of `\thispicture{...}`.

If we assume that the picture proper has been created within `gkp`'s

```

\def\gkpbeginpicture(#1,#2)(#3,#4)%
%#1, #2 xsize, ysize (dimensionless)
%#3, #4 xshift, yshift (dimensionless)
{...}
\def\gkpendpicture{...}

```

then the flexibility layer reads

```

\def\beginpicture{\bgroup
  \the\everypicture\the\thispicture
  \gkpbeginpicture(\the\xdim,\the\ydim)%
  (\the\xoffset,\the\yoffset)%
  <pictureproper>}
\def\endpicture{...}

```

A `pic.dat` database entry consists of a triple: `\lst` list element tag, `\<name>pic` the name, and a group. The group contains the defaults preceding the picture proper. The picture proper is enclosed by `\beginpicture` and `\endpicture` which also deliver the box of the size as prescribed by the parameters. Extra `\bgroup` and `\egroup` are there to keep things local.

```

\lst<name>pic{\bgroup<defaults>
\beginpicture
<pictureproper>
\endpicture\egroup}

```

Example (*Coding the database element `bintreepic`*)

The `\bintreepic` element of the `pic.dat` database reads as follows.¹⁴

```

\lst\bintreepic{\bgroup
  \unitlength.5ex\kk32
  \xoffset{-32} \yoffset{-2}%
  \xdim{66}\ydim{5}%
  \def\eertnib#1\bintree{\fi}
  \beginpicture\bintree\endpicture\egroup}
%with in the kernel blue.tex
\def\bintree{\S1\ifnum\kk=2 \eertnib\fi
  \divide\kk2 {\W{\the\kk}\bintree}%
  \E{\the\kk}\bintree}
%and accounting for the leaves
\def\eertnib#1\bintree{\fi\global\advance\k1
  \whiteS1\xytxt{ \csname\the\k\endcsname}}

```

11 Coding recursion

Long I believed that tail recursion—as used for example in the `\loop ... \repeat`—is not sufficient. We also have to cope with ‘Ackermann-like splittings’ of the recursion. For the case of the Pythagorean tree for example, this means that at each recursion level we have to take care of the left branch and the right branch, i.e. two invocations must be done. This interferes with how to handle the termination of the recursion. A problem? T_EX too difficult? Hang on there are at least two elegant and general solutions.

11.1 Recursion termination

Example (*n!*)

As pedagogical example the calculation of *n* faculty.

```
\n=... %counter var
\nfac=1 %default
\def\fac{\ifnum\n=1 \caf\fi
  \multiply\nfac\n \advance\n-1 \fac}
\def\caf#1\fac{\fi}
```

The termination is obtained because `\fac` is used as end terminator in the definition of `\caf`. `\fac` is implicitly gobbled, and the process backtracks, resulting finally in termination.

Example (Pythagorean tree)

Without explanation the Pythagorean tree recursive macro is given below.¹⁵ At each level we have two invocations—the branching—to take care of the left and right branch.

```
\def\pythtree{\ifnum\level=1 \eerthtyp\fi
  \advance\level-1
  \multiply\kk23\divide\kk32%\sqrt2
  {\leftbranch\draw\pythtree}%
  \rightbranch\draw\pythtree}
\def\eerthtyp#1\pythtree{\fi}
```

No build up of `\fi`-s, no use of `\expandafter` nor `\let`. The `\let` mechanism has been used throughout in *The T_EXbook*.

11.2 A la Knuth

In the coding of `\squine`—a quadratic spline—Knuth showed how to terminate (tail) recursions in T_EX. For `\fac` this results in the following.

```
\def\fac{\ifnum\n>1 \expandafter\dofac\fi}
\def\dofac{\multiply\nfac\n
  \advance\n-1 \fac}
```

For the Pythagorean tree the result reads as follows.

```
\def\pythtree{\ifnum\level>1
  \expandafter\dopyth\fi}
\def\dopyth{\advance\level-1
  \multiply\kk23\divide\kk32
  {\leftbranch\draw\pythtree}%
  \rightbranch\draw\pythtree}
```

Earlier I noticed the elegance of introducing an extra level, especially to circumvent too many `\expandafter`-s in a row.

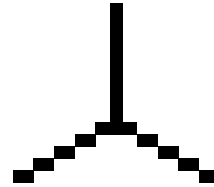
As an aside the use reads

```
\let\0\N \let\1\NE \let\2\E \let\3\SE
\let\4\S \let\5\SW \let\6\W \let\7\NW
\def\leftbranch{\advance\dir7
  \ifnum\dir>7 \advance\dir-8 \fi}%mod 8
\def\rightbranch{\advance\dir1
  \ifnum\dir>7 \advance\dir-8 \fi}%mod 8
\def\draw{\csname\the\dir\endcsname{\the\kk}}
\def\pythtree{\ifnum\level>1
  \expandafter\dopyth\fi}
\def\dopyth{\advance\level-1
  \multiply\kk23\divide\kk32
  {\leftbranch\draw\pythtree}%
  \rightbranch\draw\pythtree}
$$\unitlength0.1pt\kk128 %Size
  \level15%Order
  \N{\the\kk} %Trunk
  \pythtree$$
\tracingstats1
\bye
```

12 Trinaries

For 45° lines I used square elements. Why not use a rectangular element for 30° lines in conformance to the direction?

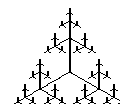
Example (Lines at 30°)



This model is obtained by

```
$$\x0pt\y0pt
  {\N{10}}{\ESE{10}}{\WSW{10}}$$
%with initializations
\linethicknesslex
\setbox\trielm=\hbox{\vrule
  width1.74\linethickness
  height\linethickness\relax}
%To account for element in 30 degrees direction
\unitlengthy\ht\trielm %default.2pt
\unitlengthx\wd\trielm %default.3482pt
\unitlength\unitlengthy%default.2pt
%and the macros
\def\WSW#1{\auxdim#1\unitlength\divide\auxdim2
  \loop\advance\auxdim-\unitlengthy
  \ifdim\auxdim>-.5\unitlengthy
    \advance\x-\unitlengthx
    \advance\y-\unitlengthy
    \xy{\vbox to0pt{\vss\copy\trielm}}%
  \repeat}
%
\def\ESE#1{\auxdim#1\unitlength\divide\auxdim2
  \loop\advance\auxdim-\unitlengthy
  \ifdim\auxdim>-.5\unitlengthy
    \advance\y-\unitlengthy
    \xy{\vbox to0pt{\vss\copy\trielm}}%
  \advance\x\unitlengthx
  \repeat}
```

Example (Trinary tree)



```
$$\x0pt\y0pt\level6 \kk128\tritree$$
%with trinary tree macro
%Pretext
$$\def\tritree{\ifnum\level=\level \eertirt\fi
  \advance\level-1 \divide\kk2
  {\N{\the\kk}\tritree}%
  {\ESE{\the\kk}\tritree}%
  {\WSW{\the\kk}\tritree}
\def\eertirt#1\tritree{\fi}
\x0pt\y0pt\level15 \kk128
\vbox to2cm{\hsize2cm
  \offinterlineskip
  \vss\tritree\vss}}$$
%Posttext
```

Remark. The `\unitlength`-s are default equal to the sides of the elementary rectangular block. The size of the tree can be controlled by `\kk`.

¹⁵Hopefully, it is self-explanatory, at least with respect to the main flow. By the way in what order is the tree drawn?

13 Epilogue

The lines at 45° have little compatibility with T_EX's rules, alas, especially with non-negligible thickness. I was surprised to realize that T_EX's defaults for rules are not symmetric around their axes in relation to the reference point.

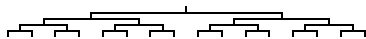
Have fun, and all the best.

14 Appendix: Binary tree and chart

Example (*Binary tree*)

```
\bluepictures\bintreepic\par
$$\bintreepic$$
```

with result



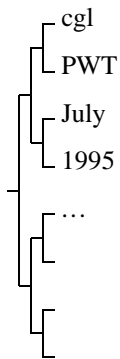
How can we add leaves¹⁶?

14.1 Rotated tree

Once we understand turtle graphics, rotating a tree can be done easily by shifting the meaning of the directions, and adjusting the positioning of the leaves.¹⁷

Example (*Rotated tree*)

```
\thispicture{\def\1{cgl}
\def\2{PWT}\def\3{July}
\def\4{1995}\def\5{\dots}
\yoffset{-16}\ydim{28}}
$$\rotatedbintreepic$$
```



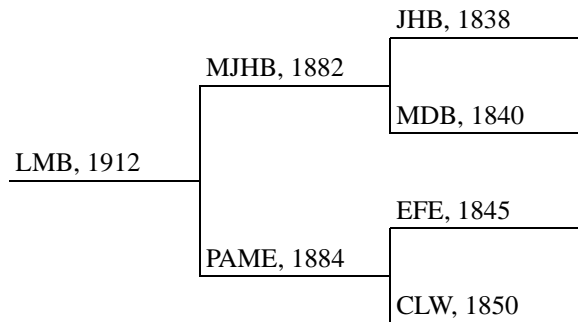
\bintree and \eertnib come with blue.tex, and \rotatedtreepic is included in pic.dat. The \rotatedtreepic entry reads as follows.

```
\lst\rotatedbintreepic{%July 1995, cgl
\bgrouper\unitlengthlex%
\let\W\N \let\exchange\E
\let\E\S \let\S\exchange
\def\1{x}\def\2{y}\def\3{a}\def\4{b}%
\def\5{piet}\def\6{hans}\def\7{etc.}%
\k0\kk16\xdim{10}\ydim{30}%
\beginpicture\bintree\endpicture\egroup
\thispicture{}}
```

14.2 Chart

Through the \bintree macro we can also obtain charts elegantly.

Example (*Chart, The T_EXbook ex22.14 248*)



obtained via

```
%labels in preorder (default in \chartpic)
\def\1{LMB, 1912}
\def\2{MJHB, 1882}\def\5{PAME, 1884}
\def\3{JHB, 1838} \def\4{MDB, 1840}
\def\6{EFE, 1845} \def\7{CLW, 1850}
\ekk8
\k0\unitlength2ex\x0pt\y0pt\kk8
\hbox{\modbintree}
%with auxiliaries
\let\Eold\E
\def\E{\global\advance\k1
\xytxt{ \csname\the\k\endcsname}\Eold}
```

Remark. An aid in finding the numbers of the branches is to delete \csname and \endcsname in \E. The way of traversal at hand is called preorder.

When using \chartpic from pic.dat the texts along the branches —\def\1{...} etc.—have to be supplied within a \thispicture, to override the defaults.

¹⁶See the PWT user guide for the answer.

¹⁷A white lie. The tree is actually mirrored because I like the leaves to be numbered from the top. In general we can rotate via POSTSCRIPT.

BLUe-2-~~L~~A~~T~~E~~X~~—expansion and some more

Kees van der Laan

Abstract

Conversion should not be a problem. It is best to use a general accepted tool, which comes with a preprint format. Until BLUe's format is generally accepted a BLUe script has to be converted to comply with publishers' formats to get it out. Conversion via T~~E~~X's expansion is exercised in this note, with as result a plain T~~E~~X Convertor Assistant, in the spirit of AWK. In netland it does not matter because BLUe's format system is available from CTAN and NTG's 4allT~~E~~X CD-ROM, and therefore everybody can format BLUe scripts, and no conversion is needed.

1 Introduction

Conversion is important and will be with us for a long time, because T~~E~~X is a fixed-point and the world around is changing. The conversion problem treated is relatively simple because it is between marked up scripts of the same kind, but nevertheless hard in its full generality.

Suppose that you have adopted BLUe's format as your personalized system. Suppose further that you submit articles to a journal—for example to MAPS or AMS. How to achieve this with as little conversion hazzle as possible? Below the solution BLUe-2-~~L~~A~~T~~E~~X~~, biased by MAPS.sty has been discussed.

To start with the bad news. I think it is undoable and a waste of energy, to provide a fully-automated conversion tool, unless there is a real demand, for example when a publisher adopts BLUe's format system.¹

The good news is that in the hands of a disciplined, knowledgeable user and restricted to canonical BLUe scripts we can get by with a little care.

I'll sum up the various ways I can think of and exercise the expansion approach via T~~E~~X. For the target I assume old ~~L~~A~~T~~E~~X~~, that is the version prior to 1994, especially to MAPS.sty. I hope,² before losing myself into details, that this note will spark discussions towards THE approach.

At the end there is the Columbus' egg solution from the point of view of authors.³

2 Why?

Of course I have touched upon the conversion problem before while submitting various BLUe's notes to MAPS. However, only after Irina Makhovaya asked me

'And what about a BLUe-2-AMS convertor?'

I started experiencing how far we can get by with T~~E~~X alone. The purpose is a 'little tool in plain T~~E~~X' to assist the conversion of a canonical BLUe script into a MAPS submission.

I don't use MAPS, TUGboat, or <YouNameIt> styles because I like to keep my scripts consistent, to use minimal markup, and because I prefer T~~E~~X, the 'fixed-point.'

3 What are the problems?

In essence we have a mapping problem between *incompatible* sets. There are incompatibilities at various levels. At the functionality level BLUe's format lacks explicit markup for the author, makes use of selective loading from databases, allows generating a ToC, and an index on the fly, and uses `\this<tag>{<options>}` and `\every<tag>{<options>}` to handle options, and the option of file verbatim inclusion in particular. Moreover BLUe's verbatims are semi-transparent, meaning an escape character is there to yield special effects. MAPS—~~L~~A~~T~~E~~X~~-biased—allows for switching from 1-column into 2-column and vice versa anywhere on the page,⁴ next to having adopted NFSS.⁵ Moreover, ~~L~~A~~T~~E~~X~~ comes with BiB~~T~~E~~X~~, and Makeindex. The difference in spelling is the least of all problems. Because of the above

I really don't see how to provide for a general, automatic and *complete* convertor.

I would classify this conversion problem as *ill-posed*.

The problem has to be simplified and confined to the tags which provide similar functionalities.

¹A script of a dozen of pages contains 50 odd different BLUe markup tags. Hand conversion takes an hour.

²My experience so far is that not many T~~E~~Xies—actually hardly none—take time for a proper discussion, alas.

³If a publisher adopts BLUe's format system, the default format, can act as a preprint style to suit the author and the publisher, with as result that an author is released from the conversion problem.

⁴It is a public secret that switching from 1-column into 2-column can't be done completely automatically.

⁵The New Font Selection Scheme.

3.1 Outer level and inner level markup

My a priori simplification is to separate outer level markup tags—which determines the look-and-feel of the typeset document—from inner level markup tags. The first are the subject of the simplified conversion problem, while the latter markup is left invariant, that is excluded from conversion.

Indispensible is a table of BLUe’s format markup tags and the corresponding tags for the journal.

Inner level markup⁶ has to do with the details of mathematics, tables and graphics, the common teasers. Look upon these as (black)boxes, with the detailed markup left at the discretion of the author, IMHO, with all respect.

Representative outer tags

BLUe script	→ MAPS.sty script
<code>\blueabstract ...</code> ...	<code>\begin{abstract}...</code> ... <code>\end{abstract}</code>
<code>\beginscript</code>	<code>\begin{document}</code> <code>\maketitle</code>
<code>\bluehead ...</code>	<code>\section{...}</code>
<code>\bitem</code> ...	<code>\begin{itemize}</code> <code>\item...</code>
<code>\bitem</code> <code>\smallbreak</code>	<code>\item...</code> <code>\end{itemize}</code>
<code>\begin quote</code> <code>\end</code>	<code>\begin{quote}</code> <code>\end{quote}</code>
<code>\ftn</code>	<code>\footnote</code>
<code>\begin</code> <code>\<escape char>end</code>	<code>\begin{verbatim}</code> <code>\end{verbatim}</code>
<code>\endscript</code>	<code>\end{document}</code>

Whatever the method applied for handling the outer level markup tags, lower level (plain \TeX) macros have to accompany the converted script for processing the inner level markup tags.

3.1.1 Prelude matter

Let me summarize some conventions of BLUe’s format prelude. In BLUe’s format system the order of specification of the prelude⁷ items is immaterial. Author and affiliation details are known by the system. References are specified in the prelude and set in a box to be pasted up at a place at will, allowing as side-effect refereeing to the literature items in one-pass.⁸ Pictures can also be specified in the prelude and selectively loaded from the `pic.dat` database. In general these must be adapted to \LaTeX ’s picture environment, or

⁶Where do outer level markup tags end and inner markup tags start? What is the borderline between them? As argued before—see the PWT users’ guide—this is context dependent, and has to be discussed casu quo agreed upon by all parties involved. Allow the use of `\btable` or Turtle Graphics. An editor must always allow lower level tags. Would you replace fancy commutative diagram markup by your markup?

⁷Also called front matter.

⁸That I don’t supply references any longer in MAPS submissions is an aside, which does not matter for the general problem. I urge readers to browse separately `lit.dat` by keyword, to spot the details of the references.

⁹BLUe’s format boxes prelude elements. To move the abstract part I have to borrow a little more from BLUe’s format system.

¹⁰IMHO, the question is to what level?

perhaps recasted into POSTSCRIPT. Acknowledgements is something which does not belong to the script proper, similar for keywords and abstract, IMHO, with all respect. The non-proper script issues have to be supplied in the prelude part in BLUe’s format system, but if you don’t that is fine too.

All the above aspects differ with MAPS style—especially with \LaTeX which is underneath—in philosophy or markup conventions, and usually both.

Moving document elements such as the abstract part after the `\maketitle` is considered as fine-tunings.⁹

3.2 MAPS submission

Jos Winnink—one of MAPS editors—commented that the requirements for MAPS are that scripts should *not*

- exercise their own layout¹⁰
- interfere with existing control sequences; `\globals` especially are nasty.

MAPS processed scripts marked up in *old* \LaTeX up till 1995.

3.3 Canonical BLUe scripts

Plain \TeX contains $O(1K)$ control symbols and sequences. BLUe’s format adds $O(.1K)$ to that. My ‘little’ convertor does not account for all those. With a canonical script I mean a minimal marked up script, restricted to the use of the common high-level `blue.tex` markup tags—the so-called FUTs Frequently Used tags—as enumerated in the Appendix ‘Canonical BLUe Tags.’ The bonus of this disciplined markup is a clean script, hardly inflected by markup. Conversion of this class of scripts shouldn’t halt and shouldn’t leave you with incomprehensible error messages.

\TeX is extensible. An author can define special purpose lower level macros with names I can’t foresee. To account for these I have provided a `toks` variable `\invariantconvertor` with the purpose that the provided tags won’t be expanded during conversion, that is remain invariant.

A BLUe script lacks explicit markup for the author because it is a personalized system and therefore BLUe knows its author. The convertor must be assisted by prompting the author. An example of prompts which should be supplied in the file `aidsconvertor.tex` reads as follows.

```
\authorconvertor{<YourName>}
\invariantconvertor{\<tagone>\<tagtwo>...
  \<tagn>}
```

Mark up for affiliation in \LaTeX is included in `\author`.

3.4 And what about the active parts?

Active documents are all about things happening behind the scenes, like automatic distilling information from the static elements, to be used for example in headers, footers, table of contents, and index. For the moment I consider these as fine-tunings. Tokens like `\loadtocmacros` and `ilks` are suppressed during conversion.

Cross references occur within math formulas via `\ref` and `\crsref`. Optionally `\ref` is followed by a symbolic name. For the moment this name has to be supplied in the invariant set. The cross reference macros from BLUe's format have to accompany the converted script.

4 The challenge

Given the above discussed complexities and restrictions the challenge is to develop a trustworthy Converter Assistant which works the way you expect.

Coding the convertor in plain T~~E~~X is a non-trivial exercise in macro writing.

5 Conversion approaches

Stripped from incompatibilities the conversion has been reduced to transforming the various begin and end tags, blue headings, quotations, loops, if clauses, verbatims and `ilks`.¹¹

The possible ways I can think of are

- a Demarkup a BLUe script and insert the publisher's markup tags
- b Leave the script as is, but provide
 - 1 `\maps` command, to yield the MAPS look-and-feel via plain T~~E~~X
 - 2 macros which invoke the MAPS markup commands via L~~A~~T~~E~~X
- c Convert BLUe's outer level markup tags into MAPS markup tags, via
 - 1 an editor
 - 2 an AWK, Perl, or ...script¹²
 - 3 T~~E~~X itself; the conversion is done via expansion of `\write`, mainly.

5.1 Method a: demarkup first

Roughly speaking method a is in use for example while transforming a Wordperfect script into (L~~a~~)T~~E~~X.¹³ To demarkup a script is very close to tag replacement and therefore this approach is close to method c1. The advantage is that it works at the expense of some tedious editing and trial-and-error formatting runs. When (minimal) mark is in-

serted in the final stages of the document preparation process this method is feasible.

5.2 Method b1: maps format

This method has all to do with acceptance of BLUe's format. The function of `\maps` is similar to say `\report`, but with as result of the BLUe script a document with the MAPS look-and-feel, processed by plain T~~E~~X. Similar to `\maps` one could provide `\ams`, or `\YouNameIt`.¹⁴ Typical for this approach is that for example `\bluehead` remains as is, but `\prehead`, `\posthead`, and `\headfont` are adapted. Because MAPS editors have adopted to process MAPS completely via L~~A~~T~~E~~X, it is unlikely that `\maps` will ever see the light of day.

5.3 Method b2: BLUe as user interface

The idea is that `blue.tex` is seen as a user interface to L~~A~~T~~E~~X. This method redefines BLUe tags with as replacement text invokes of MAPS commands. A prototype `\bluemaps` for BLUe script markup tags to expand into the L~~A~~T~~E~~X-biased MAPS tags. This method respects MAPS to be processed completely by L~~A~~T~~E~~X.

Example (*Representative definitions*)

```
\def\bluehead#1\par{\section{#1}}
%
\def\bitem{\bgroup\begin{itemize}%
  \let\bitem\item
  \def\smallbreak{\end{itemize}\egroup}%
  \item}
```

5.4 Method c1: hand replacement

Little knowledge is required. Just a lot of work each time. Single shot contributions are handled this way in practice.

5.5 Method c2: converter scripts

I suspect MAPS editors to use method c2 when I submitted notes together with the `bluemaps` macros. The advantage of their approach is that they don't end up with a mess of macro sets.

5.6 Method c3: conversion via expansion

This is an unusual approach, and induced by T~~E~~X's mouth processing capabilities.¹⁵ If it is true that T~~E~~X is all about text replacement this approach deserves attention, if not for finding out its limitations.

¹¹However, in practice I stumbled upon another difference in style. BLUe's format system allows for running-in heads which is absent in MAPS as such, yielding not nice opening sentences when conversion is done more or less automatically. I like running-in heads especially for `\subsubhead`. The way out is to replace these `\subsubheads` by `\(sub(sub))paragraphs`.

¹²Not further elaborated here, because of lack of hands-on for these tools. Maybe, I'll come back on this.

¹³I know of automatic convertors, and I'm also aware of their limitations. The activity at the moment is to start from the `dvi` file, to provide a `dvi2<YouNameIt>`, similar to `dvi2ps` and such like.

¹⁴As volunteer this development work is not so interesting for me. An organization can hire me to do the job.

¹⁵Bernd Raichle communicated that he has already applied the basic idea in his CTAN convertor programs: `tex-archive/support/convert/...`

5.7 Method x: conversion via modified T_EX

At TUG 95 Sebastian Rahtz has reported about the Elsevier project L^AT_EX-2-SGML. They envision to modify T_EX such that an appropriate dvi file will emerge, an ‘intermediate,’ to be processed further. The method builds upon the T_EX parser.

Given their mass ‘audience’ a foolproof, cost-effective tool is needed. My convertor is personalized, small scale, and expects a little TLC, Tender Loving Care.

However, one never can tell what future has in store for us.

6 Conversion via expansion

The basic idea of the method elaborated in this note is to read and write each line¹⁶ with the conversion done while writing. The convertor is table-driven. This basic loop reads in its simplest form as follows.

```
\loop\ifeof\bluesrc \break\fi
  \read\bluesrc to\inp
  \expandafter\lop\inp\eof
  \immediate\write\maps{\inp}
\pool
%with auxiliaries
\def\loop#1\pool{#1\loop#1\pool}
\def\break#1\pool{\fi}
```

Crucial is the inspection of `\inp`. The *first* token is lopped off and depending on its value non-expandable actions are performed, for example redefining `\inp`.

Let us discuss representative situations, for example how to handle BLUe’s `\blueabstract`, `\bluepictures`, `\beginscript` and `\endscript`, `\bluehead`, `\bitems`, `\ftn`, `\beginverbatim` and closing, comments, and `\this<tag>s`.

Some of the problems are induced by BLUe’s minimal markup spirit,¹⁷ such as explicit opening tags and implicit closing tags and as few curly braces as possible. This opposed to as well explicit opening as closing tags and to surround arguments by braces. Because conversion via expansion is unusual I have exercised it a little, to illustrate the idea and to stirr up discussions.¹⁸

Disclaimer. Beware there is no complete solution, no fine-tunings, no general production tool, and definitely no fool-proof version as yet. (In-line) Verbatims have to be verified for sure. However, one never can tell whether the used T_EXniques will turn out to be useful some day.

6.1 Abstract

The minimal markup in a BLUe script is to precede the abstract by `\blueabstract` and end it by the implicit markup of a blank line. The abstract text can extend several lines. The convertor encloses the abstract text by `\begin{abstract}` and `\end{abstract}`. `\bluekeywords` goes similar.

Restricted to the handling of `\blueabstract` only the coding reads as follows.

```
\def\blueabstract#1{\begingroup\abstractcnt1
  \def\inp{\string\begin{abstract}\gobble#1}}
\def\par{\ifnum\abstractcnt>0 \endgroup
  \def\inp{\string\end{abstract}}\fi}
```

`\inp` contains the line read. The first token is `\blueabstract` and the rest of the line is supplied as argument to the invoke of `\blueabstract` by the main processing loop. The blank line yields `\inp{\par}` and `\par` is also included in `\convertorset` meaning that ‘`\par`’ is executed, that is the above replacement text is executed.

6.2 Pictures

The codings as such are already separated from the script and stored in `pic.dat`. This `\thispicture...` is left invariant. The picture names should be added to `\invariantconvertor`. The same holds for `\bluereferences`. The macros should accompany the converted script.

6.3 Beginscript and endscript

The conversion of `\beginscript` goes via the following redefinition. Note that L^AT_EX needs an explicit `\maketitle`. BLUe sets the title automatically when `\beginscript` is encountered.

```
\def\beginscript{\string\begin{document}
  \string\maketitle}
```

`\endscript` is similar in principle. A little more has to be done, however, to suppress the `\par` at the end of the file. When `\endscript` is encountered, `\end{document}` is written to the file and the conversion is stopped.¹⁹

6.4 Headings

The problem is to recognize the situation, to get hold of the header text—the argument—and to insert the latter within braces after `\section`. I assume that the header text is a 1-liner, and that the markup tag is the first tag on the line. A nuisance is to get rid of inserted spaces.²⁰

```
\def\bluehead#1{\def\inp{%
  \string\section{\gobble#1}}}
```

Similar to headings is `\blueexample`. A macro `\example` must accompany the converted script because L^AT_EX lacks `\example`.

¹⁶Do remember that either a line is read, or more than one line until the braces match.

¹⁷In pursuit of Knuth.

¹⁸Have a glance at the included toy example to get an idea of what can be handled automa*g*ically, already.

¹⁹Be aware of the difference between `\string` and `\noexpand`. The latter inserts a space after the non-expanded token.

²⁰The inserted space before the closing brace does not harm most of the time. `\example` must be defined appropriately, however.

6.5 Items

First of all there are the following varieties in BLUe: `\item`, `\bitem`, `\nitem`, `\aitem`, and `\Aitem`. BLUe’s format assumes that a sequence of item tags is terminated by `\smallbreak`.

L^AT_EX provides the bulleted, enumerated and descipited items, all surrounded by proper opening and closing tags.

The `\bitems` have been treated in the example below, to show one way of doing. I consider the handling of the `*` option of L^AT_EX’s sectioning tags as fine-tunings. `\input` goes similar. Restricted to the case of `\bluehead` only the coding reads as follows. `\remove` removes its argument from `\setconvertor` with the effect that in the main processing loop `\bitem` is treated differently for the second, third etc. time, untill the group is closed that is when `\smallbreak` is encountered.

```
\def\bitem#1{\begingroup
  \def\bitem{\string\item}
  \immediate\write\bluemaps
    {\string\begin{itemize}}
  \advance\bitemcnt1 \remove\bitem}
\def\smallbreak{\ifnum\bitemcnt>0
  \endgroup
  \def\inp{\string\end{itemize}}\fi}
```

The `\items` go similar. The redefinition of `\inp` must insert square brackets around the argument of the `\item`. The `\nitems` are slightly different. Nested items have been accounted for too. `\aitems` and `\Aitems` are considered as fine-tunings.

6.6 Footnotes

As such there are no problems. However, the footnote text is read completely—remind the The T_EXbook217 ‘Additional lines are read if necessary, until an equal number of left and right braces has been found.’

L^AT_EX won’t bark but the converted script might contain long lines.

6.7 Verbatim tags

`\thisverbatim`s are preceded by `%` because they are incompatible. However, the contents is copied into `\thisverbatimconvertor` and used during the conversion.²¹

This enables flexibility with respect to the escape character which precedes the `endverbatim`. `\everyverbatim`s are just preceded by `%`.

`\beginverbatim` token is redefined straightforwardly. The exclamation mark `!` is the default escape character to end a verbatim. Its catcode is set to 0. The catcodes of the braces are changed in order to process the verbatim line-by-line.

The in-line verbatims need (pre)insertion of `\verb`. Because these can occur on any position of the line no catcodes

are changed. So the material inside is expanded during conversion.

6.8 Verbatim texts

Handling verbatim opening and closing tags is the top of the iceberg. The verbatim text must be handled appropriately. Like in ordinary verbatim we have to switch off temporarily the backslash as escape character.²² A beneficial side-effect is that T_EX does not continue reading lines until braces match. Verbatim texts after `\begindemo` and Knuth’s demos are handled similarly.

6.9 Comments

The catcode of the comment symbol, `%`, is set to 12, with as result that comment lines won’t disappear. Be aware that tokens in there are expanded.

6.10 Explicit skips

I decided to replace `\bigskip` and ilks by a blank line. It is descriptive markup we are after, aren’t we?

6.11 Newcounts and such like

It can happen that counters and such like will be used in the script. Not only do we have to take care of `\newcount`—an outer `def`—but also of the counter name, which can be an arbitrary control sequence.

```
\def\newcount#1{\def#1{\string#1}\def\inp
  {\ea\string\csname newcount\endcsname#1}}
```

It is assumed that each `\newcount\<countername>` occurs as such on a line.

I chose to neglect L^AT_EX’s `\newcounter`.

Remark. It is tempting to treat `defs` in the same way. I refrained because `defs` can occur more than once with the same name for example `\def\data`.

6.12 Facts from real life

In converting a `\btable` and its data, I stumbled upon that read does not per se read a line but, if the case arises, it reads up to and including the line with the closing brace.²³

6.13 Conversion table

The table consists of redefinitions of BLUe’s format tags. First the simple ones which when expanded yield the required substitution. Next there are definitions which should remain invariant, but should not be expanded during the write. Third there are (empty) definitions which omit the tag during the write. Finally, we have headings and ilks which next to changing the tag name should insert braces *around* the head text. Below representative parts of the table have been selected to convey the idea.

```
%Conversion table
%1. Cs-s which redefine \inp, or do
% some other inexpandable process.
%The cs-s are also collected in the
```

²¹This implies that `\catcode` isn’t converted as such.

²²Still a problem with in-line verbatims.

²³If we wish to keep the layout invariant we could change the catcodes of the braces before reading a line. However, this conflicts with other situations, like the argument of an `\item`.

```
%set \setconvertor for identification.
\def\bitem#1{\begingroup
  \def\bitem{\string\item}
  \immediate\write\bluemaps
    {\string\begin{itemize}}
  \advance\bitemcnt1 \remove\bitem}
%...
\def\yields{\endgroup\def\inp{!yields}}
%2. Cs-s which expand irregularly
\def\beginscript{\string\begin{document}
  \string\maketitle}
%...
\def\ftn{\string\footnote}
%3. CS-s which should disappear
\long\def\process#1{\let#1\empty}
\expandafter\fifo\disappearset\ofif
%4. CS-s which should remain invariant
\long\def\process#1{\def#1{\string#1}}
\expandafter\fifo\invariantset\ofif
%5. To be commented out
{\catcode\%=12\gdef\percent{}}
\long\def\process#1{\def#1{\percent\string#1}}
\expandafter\fifo\commentset\ofif
%end conversion table
```

6.14 Toy test

A toy test program to demonstrate representative cases.

```
\bluetitle BLUe-2-MAPS
\bluesubtitle expansion and some more
\blueabstract BLUe-2-MAPS is all
  about conversion.
\newcount\cglcnt
\beginscript
\bluehead head text
Text
\bitem first bitem
\itemitem- first subitem
\bitem second bitem
\smallbreak
\item- first item
\item- second item
\smallbreak
\nitem first nitem
\nitem second nitem
\itemitem- first subitem
\itemitem- second subitem
\smallbreak
\item- first item
\itemitem- first subitem
\itemitem- second subitem
\item- second item
\smallbreak
\begindisplay
...&... \cr
\enddisplay
Math
$$a^2+b^2=c^2\ref\pyth$$
A famous truth from antiquity \crsref\pyth.
\thisverbatim{\catcode\|=12 }
\beginverbatim
  \def\cs#1{{\tt\char\| #1}}
  \beginscript
  \bluehead blah-blah
  ...& $ | ^ _ %
\endscript
!endverbatim
```

```
\begindemo
...
!yields
...
\enddemo
\loop
...
\ifnum\cglcnt=0
...
\advance\cglcnt-1
\repeat
\mycs{abc}
\def\htablecaption{\leftarrow\hfill
  C a p t i o n\hfill\rightarrow}
\def\first{\hbox to6ex{\hss First\hss}}
\def\header{\hbox to18ex{\hss Header\hss}}
\def\rowstblst{{\hfil$\vcenter to20ex{\vss
\offinterlineskip
\fifo rowstblst\ofif\vss}$}}
\def\process#1{\hbox to0pt{\hss
#1\hss}\kern.5ex}
\def\data{\vcenter{\offinterlineskip
\hbox{Table}\kern1ex\hbox{Proper}}}$}
\def\footer{\leftarrow\hfill\small
  F o o t e r}\hfill\hfill\hfill
  $\rightarrow}
$$\vcenter{\htable\data}$$
\endscript
```

6.15 User guidance: aidsconvertor.tex

The above example was converted with the following aidsconvertor.tex.

```
%Auxiliary for BLUe-2-MAPS: aidsconvertor.tex
%Personalized info
\authorconvertor{Kees van der Laan}
%To assist conversion
\invariantconvertor{\mycs\a\b\pyth}
```

Other token variables available are

\commentconvertor to precede the tags by comment symbol, \deleteconvertor to neglect the tags.

6.16 Test result

```
\documentstyle{article}
\author{Kees van der Laan}
\input{bluemaps.tex}
\input{bluemaps.xtr}
\title{BLUe-2-MAPS }
%\subtitle{expansion and some more }
\begin{abstract}BLUe-2-MAPS is all
  about conversion.
\end{abstract}
\newcount\cglcnt
\begin{document} \maketitle
\section{head text }
Text
\begin{itemize}
\item first bitem
\begin{description}
\item[-] first subitem
\end{description}
\item second bitem
\end{itemize}
\begin{description}
\item[-] first item
```

```

\item[-] second item
\end{description}
\begin{enumerate}
\item first nitem
\item second nitem
\begin{description}
\item[-] first subitem
\item[-] second subitem
\end{description}
\end{enumerate}
\begin{description}
\item[-] first item
\begin{description}
\item[-] first subitem
\item[-] second subitem
\end{description}
\item[-] second item
\end{description}
\begin{display}
...&... \cr
\end{display}
Math

$$a^2+b^2=c^2$$

A famous truth from antiquity \crsref{pyth.}

\begin{verbatim}
\def\cs#1{{\tt\char'\#1}}
\begin{script}
\buehead blah-blah

...& $ | ^ _ %
\end{script}
\end{verbatim}%the space is a kludge

\begindemo
...
!yields
...
\enddemo

\loop
...
\ifnum\cglcnt=0
...
\advance\cglcnt-1
\repeat

\mycs{abc}

\def\btablecaption{ $\leftarrow$ \hfill
Caption\hfill $\rightarrow$ }
\def\first{\hbox to6ex{\hss First\hss}}
\def\header{\hbox to18ex{\hss Header\hss}}
\def\rowstblst{{\hfil$\vcenter to20ex
{\vss\offinterlineskip\ifrowstblst\ofif
\vss}$}}
\def\process#1{\hbox to0pt{\hss
#1\hss}\kern .5ex}
\def\data{ $\vcenter{\offinterlineskip
\hbox{Table}\kern 1ex\hbox{Proper}}}$ }
\def\footer{ $\leftarrow$ \hfill\small
Footer\hfill\hfill\hfill $\rightarrow$ }
 $\vcenter{\btable\data}$ 
\def\data{1\cs2\rs...\cs...\rs}
 $\btable\data$ 
\end{document}

```

Remarks. I don't know how to cope with nested verbatims in L^AT_EX, so I inserted a space to fool the pattern matching.

The last lines about information for `\btable` have been edited for 2-column format. T_EX reads either a line or

more than one line to balance explicit braces. In general I would suggest to take this part out and include it in `bluemacs.xtr` under an appropriate name.

7 Converter Assistant Program

The considerations in the beginning of this note led to a prototype. After a couple of months I returned to the subject and pushed it to its current state.

7.1 Design

A to be converted script has the following classes of markup tags, which should

- do more than expansion
- be expanded into
 - the control sequence itself
 - nothing
 - the control sequence itself preceded by %.

The documentation in this note was developed simultaneously,²⁴ and functioned also as design specifications.

In working with the Converter Assistant fine-tunings had to be accounted for.

7.2 Coding

The main loop consists of processing the input line-by-line. Each line is split in two parts: the first token and the rest. The first token is checked whether it is in the set of tokens which do more than expansion.

The control sequences take the suffix `converter`.²⁵ The educational pieces of code enclosed in this note have not been cluttered by the suffix.

User guidance is expected in the file `aidsconverter.tex`.

7.3 Availability

The Converter Assistant, this note as well as the kernel `bluemacs.tex` can be obtained from the CTAN.

7.4 Robustness

The converter is not robust It will go astray when control sequences which do more than expansion during conversion are used in positions different from the first of a line. In short use THE markup tags only in the first position of a line.

8 bluemacs.tex accompanying macros

A converted script is accompanied by lower level macros which have been left invariant and can better be processed by this suite.

The kernel `bluemacs.tex` contains macros for

- `btable`
- Pascal fragments
- cross-referencing for math

²⁴My PM-LP—Poor Man's Literate Programming—way of working.

²⁵The control sequences of the program itself should not occur in the scripts.

- turtle graphics, and
- Knuth's gkppic 'picture' subset.

Next to this kernel I supply extra macros `bluemaps.xtr` specific for the document at hand.

9 Push the button

There are two phases: running the CA via plain \TeX , and verifying the result by \LaTeX .

9.1 Convector Assistant processing

Alas, the use of the CA is not that simple. Some preprocessing and postprocessing has to be done. The preprocessing consists of

- verify whether the used markup tags belong to the FUT, Frequently Used Tags (see the Appendix), and whether they occur on the first position of a line
- eliminate catcode changes, except for those in `\thisverbatim`
- eliminate Index Reminder markup
- assign control sequences which should remain invariant to `\invariantconvector`, and similarly for `\deleteconvector` and `\commentconvector`
- extract detailed markup such as needed for a `\btable` from the script and accumulate this in `<jobname>.xtr`²⁶
- look for use of ! (escape character in the convector) and adapt this
- change markup for space after control sequences into `\cs{}_`
- remove `\input blue.tex`, `\loadtocmacros`, `\loadindexmacros`, `\pasteuptoc`, `\sortindex`, `\pasteupindex`, and the overfull circumventing `\tolerance=500` `\hbadness=499` `\hfuzz=5pt`²⁷ and such like
- move the keywords and abstract part after `\beginscript` if they are supported by the style file
- make fileverbatims harmless by removing `\input` from `\thisverbatim`.²⁸

Postprocessing has all to do with verifying the result and supplying pictures and other macros needed additional to `bluemaps.tex` to be sent along with the converted script. I called this extra file `bluemaps.xtr`. Tedious it is and beware name clashes.

While the CA is running keep an eye on the transformed file, and invariantly you will stumble upon unknown control sequences. Don't panic, just go on, and add these to the toks variable `\invariantconvector`.

²⁶or enclose these (temporarily) by `\beginverbatim` and `(escapechar)endverbatim`

²⁷See Phil Taylor's Pragmatic TUGboat article on the issue.

²⁸`\bluefileverbatim` is in the comment set.

²⁹This is after conversion no longer appropriate, so why not just delete it as part of preprocessing phase?

³⁰The latter should not be a too difficult job, because BLUe's format has been setup in a modular way, as a compatible extension of plain \TeX , and `manmac`, the macros used by Knuth himself.

9.2 Problems left

Whenever I stumbled upon something I could not solve I could get around it after a night of sleep. For example testing for `\input blue.tex` and changing this into `\input{bluemaps.tex}`. The Columbus's egg solution I adopted was to let the author delete `\input blue.tex`²⁹ and let the Convector Assistant automatically insert `\input{bluemaps.tex}`. I also decided to let the CA automatically insert `\input{bluemaps.xtr}`, the default extra lower-level macros.

10 Columbus' egg

What if ... publishers adopt BLUe's format system with for example the default format as the preprint format?

Aha, then we don't have to muddle around any longer, because the author and the publisher use the same tags. The publisher is free to adjust the tags to reflect the layout of the various journals, in short provide in-house production versions of the '`\maps`' tag.³⁰

As far as I can see it, this is the way how publishers and authors cooperate nowadays with \LaTeX as formatting tool.

In netland—especially as *self-publishing author*—the problem is non-existent, because BLUe's format system is available from CTAN and NTG's 4All \TeX CD-ROM. Everybody can format BLUe scripts, and conversion is not needed.

11 Acknowledgements

Thank you Irina Makhovaya for challenging me to consider conversion of BLUe-2-`<YouNameIt>`. As usual Jos Winnink lend a helping hand in the finishing touch for MAPS inclusion.

12 Conclusion

Aren't we captivated by \TeX nical solutions, while now and then (part of) the problem must be solved by other means? IMHO, with all respect, conversion is such a kind of problem. We should not convert but a `\maps`, `\ams`, or `\<YouNameIt>` tag should be provided, as variants on `\report`. The assumption is that general accepted tools or formats—standards so to say—have to be used. In short method `b1` is the way to go.

Necessary conditions for general acceptance of a \TeX format by users are

easy to use, well-documented, extensible.

BLUe's format system fulfills the necessary conditions, IMHO. I hope that `blue.tex` will enjoy general acceptance, to start in off-off- \TeX netland.

Conversion via T~~E~~X's expansion—or an AWK, Perl, ... script—in short method c, is what I do in want for acceptance of BLUe's format by the MAPS editorial board. (MAPS) editors can forget about the format in which the document originated.

The toy example has shown the feasibility of the approach. It needs the Tender Loving Care of a knowledgeable user.

An early version of this note was converted with the following `aidsconvertor.tex`

```
%Personalized info
\authorconvertor{Kees van der Laan}
%To assist conversion
\invariantconvertor{\mycs\a\b\cglcnt
  \pyth\shiftright\this\every}
```

Conversion is an important issue, it deserves attention. Fine-tunings of the convertor will emerge if the need arises.

A real-life example would be BLUe-2-AMSPPT. I would embark such a project only with approval and support of the AMS.

My Convertor Assistant taught me to markup my scripts simple, consistent, disciplined and above all minimal.

I expect my CA to become more and more useful because it will *learn* on-the-fly what T~~E~~X lingo I use.

My case rest. Have fun, and all the best.

13 Appendix: Canonical BLUe Tags

The following markup tags have been accounted for in the Plain T~~E~~X Convertor Assistant. These main tags—also called FUTs—have been borrowed from the PWT guide Appendix Formats. In the following table the tags which take a ' as suffix have been used in this script.

```
Preliminary matter
\bluetitle'
\bluesubtitle'
\bluekeywords'
\blueabstract'
\blueissue'
\bluereferences
\bluepictures
\input'
```

```
Copy Proper
\beginscript \endscript'
\bluehead'
\thissubhead{...}
\bluesubhead'
\thissubsubhead{...}
\bluesubsubhead'

Inner level
\begindemo...<escapechar>yields...\enddemo
\everyverbatim{...}'
\thisverbatim{...}'
\beginverbatim'
<escapechar>endverbatim'
(and in-line \vrt\thinspace...\thinspace\vrt)'
\beginquote'
\endquote'
\em'
\dash, \Dash'
\dots'
\oldstyle'
\begindisplay
\enddisplay
(with & and \cr)
\displaycenterline{...}
\begincenterdisplay
\endcenterdisplay
(with & and \cr)
\htable \beginhtable \endhtable'
(with \data{...\cs...\rs...\cs...\rs...}
  \header{...}
  \footer{...}
  \rowstblst{...}
  \vruled{...} )'
\ref<name>
\crsref<name>
\ftn{...}'
\item{...}'
\itemitem{...}'
\aitem %to be fine-tuned
\Aitem %to be fine-tuned
\bitem
\nitem
\blueexample'
\thispicture{...}
(with \unitlength=...)
\cs{<tagname>}'

Back matter
\pasteupindex
\pasteupreferences
\pasteuptoc'
\sortindex
```

Graphics and T_EX—a reappraisal of METAFONT/MetaPost

Kees van der Laan

Abstract

It is all about the author's first steps in METAFONT, for creating graphics to be included in T_EX documents, with a wink to MetaPost. The graphics comprises 2D pictures and 2.5D images of 3D objects via projection techniques. Learning METAFONT was much easier than learning T_EX. Included examples are: cat, Hilbert curve, and Gabo's linear construction in space no 2. A few highlights on macro writing in METAFONT have been selected. The appendix contains the table of contents of my anthology.mf file of examples.

1 Introduction

The handling of graphics in T_EX scripts has a history.¹ There are three approaches from the document preparation point of view

- T_EX alone
- T_EX and METAFONT
- Use of 'third party' graphics tools.

Everybody uses epsf—encapsulated POSTSCRIPT—in the last two cases as medium to merge the graphics with the T_EX script in order to get the results out.²

1.1 T_EX alone

L^AT_EX's picture environment is the common example for this class, though plain T_EXies might use the macros from Graham/Knuth/Pastashnik,³ which reflect a subset of L^AT_EX's picture functionality in plain T_EX. In BLUe's format system I also introduced the use of 'Turtle graphics.' I have used this in PWT—Publishing with T_EX user's guide—for simple fractals.⁴

Interesting too is Gurari's approach.

In scientific circles a problem is how to paste up mathematical graphs electronically. An approach is to calculate the graphs via Pascal or so and let Pascal generate the T_EX code for the graph. A few years ago I shuffled and typeset bridge hands via this method.

These methods will not be dealt with in this paper.

1.2 T_EX and METAFONT

John Hobby recognized the power of METAFONT for the design of (systematic) graphics and married POSTSCRIPT's outlines to METAFONT in his MetaPost, banning the

bitmap approach.⁵ This is the path I'm on to emulate Naum Gabo's constructive art.

Other approaches are for example

- Jackowski's⁶ mftoops package to transform METAFONT files into POSTSCRIPT and vice versa
- Leathrum/Tobin's mfpic which applies METAFONT's character handling technique to export graphics in general.

Graphics via METAFONT is the main subject of this paper.

1.3 Use of third party tools

Of late more and more sophisticated graphics and multimedia software emerges, which allow *interactive* graphics⁷ among other things. Happily, import and export of POSTSCRIPT files is possible, and therefore the software—Adobe Illustrator, Photoshop, Coreldraw, etc.—can cooperate with T_EX and METAFONT.

Of course one could use POSTSCRIPT throughout.

All of the approaches have their pros-and-cons. What to use—and when—depends as usual on your circumstances.

The third party tools will not be dealt with in this paper.

2 METAFONT

Learning METAFONT was easier for me than to learn T_EX. I picked up the flavour from Knuth's (first) book in the field 'T_EX and METAFONT—new directions in typesetting.' Next, I read METAFONT Book to absorb the ideas, possibilities and details. Finally, and inevitably, I exercised graphics examples borrowed from literature.⁸

¹In the old 'mainframe' days documents were prepared with space left open for graphics and tables, prepared by other tools, to be pasted in.

²Alas, there is no standard as yet for the use of \special-s. I hope Rockiki will succeed with his proposed standard. The inclusion of epsf in (La)T_EX documents will not be treated in this paper.

³Used for typesetting their book Concrete Mathematics.

⁴In this note the Hilbert curves of order 1 and 2 have been handled via 'Turtle graphics.'

⁵Designed under UNIX but also ported to DOS. AT&T has released MetaPost and add-ons in the public domain. Thank you.

⁶And friends.

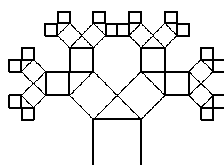
⁷As opposed to systematical, reproducible, declarative graphics.

⁸This resulted in a file called anthology.mf, which I can easily walk through on my Mac with BLUe Sky's METAFont. It is not (yet) a database to load selectively from.

A superb survey of the language and what you can achieve with it, is given by Hobby in his ‘A user manual for MetaPost, AT&T, CSTR 162.’⁹

Next to that there are the introductions: Tobin’s ‘METAFONT for Beginners’ and Jackowski’s GUST tutorial. The latter, alas, is in Polish.

During my building up of ‘A graphics anthology in METAFONT,’ I found the path data structure an eye-opener. It shed new light on algorithms for the drawing of Hilbert curves, Sierpiński curves, Pythagorean trees, and ilks, which are formulated usually recursively.



However, use of the path data structure in combination with METAFONT’s operation on pictures, like `addto ...`, yields elegant, concise and fast non-recursive programs.¹⁰

2.1 MetaPost and extensions

MetaPost is (nearly) upward compatible with METAFONT, and concentrates on graphics. It has banned the bitmap approach and combines the goodies of POSTSCRIPT with METAFONT. MetaPost also provides for integration of text and graphics, next to suitable I/O.

Hobby’s graph extension has been treated in AT&T CSTR 164. It is all about typesetting scientific graphs, the functionality of troff’s `grap` recasted in MetaPost. The following functionalities are provided¹¹

- automatic scaling
- automatic generation and labelling of tick marks or grid lines
- multiple coordinate systems
- linear and logarithmic scales
- separate data files
- ability to handle numbers outside the usual range
- arbitrary plotting symbols
- drawing, filling and labelling commands for graphs.

3 Why?

I have a keen interest in the works of Naum Gabo.¹² When I heard of METAFONT it occurred to me that I could em-

ulate his works. To put it in another way. I was curious whether METAFONT could be used conveniently as a *design* tool for 3D objects too. From a computer science point of view Gabo’s sculptures are very interesting, especially those composed of regular surfaces.¹³

4D impression can be obtained from 1D information via the use of regular surfaces, projection techniques, and varying viewpoint angles.

I guess, if Gabo would have lived now, he would have exploited the use of computers, because programming a computer—professionally known as software engineering—is a constructivistic activity.

3.1 POSTSCRIPT straightaway

I have no convincing arguments against nor in favour. Perhaps, it is a matter of taste, biased by the stability and high-levelness of METAFONT, next to the reputation of Don Knuth. On the other hand POSTSCRIPT is a de facto standard, POSTSCRIPT can be included in (La) \TeX documents, and for the moment POSTSCRIPT is intermediate in the chain `dvi`→`ps`→`PDF`,¹⁴ and therefore a nodding knowledge of POSTSCRIPT is beneficial anyhow.

One thing for sure, however. Because of METAFONT’s path data structure I uncovered a new coding for the Hilbert and similar curves, next to a systematic derecursion technique.

The question evaporates into thin air if we look at MetaPost as POSTSCRIPT with METAFONT as user interface. Hobby has enriched MetaPost by the add-on tool `graph`, which provides troff’s `grap` functionality and a little more.

4 Examples

In the sequel a few excerpts from my METAFONT anthology.

The included codes work as such on a Mac with BLUE Sky’s METAFONT. How to code the pictures, unblurred by the ‘shipit’ details, was the purpose.

When the METAFONT shipping out of characters is used the code must be adapted, such as enclosing it by `beginchar` and `endchar` and providing `beginchar` with the appropriate arguments as treated in METAFONT Book.

For using MetaPost a few adaptations are needed such as deleting the bitmap operations `cullit`, `screenstrokes`, and `ilks`, and enclosing the picture by `beginfig` and `endfig`, or `begingraph` and

⁹Can be obtained by mailing to `netlib@research.att.com` send 162 from `research/cstr`. CSTR 164 can be obtained analogously.

¹⁰Wirth has discussed the trade-off between data structures and algorithms, in his *A + DS = Programs*. Apparently, he did not think of the path datastructure at the time.

¹¹Borrowed from the CSTR 164 Introduction.

¹²Born Pevsner at Brjansk, 1890.

¹³A regular surface is determined by its boundary—1D—of which points are connected by straight lines. It conveys a picture of a 3D object.

¹⁴Adobe’s Portable Document Format, which abstracts from the preparation tools and concentrates on the userside—the consumer—by the concise PDF format and the Acrobat reader, distiller (cross-referencing), exchange and ... multi-media tools.

endgraph when the graph extension is used. The effect of reverse video via

```
addto blackbackground also -blackpicture
```

has to be adapted too, for example via the use of (white) colour.

4.1 Cat

This example is all about the use of a pen of varying width. It gives an impression of what can be attained by METAFONT/MetaPost with respect to classical drawing.

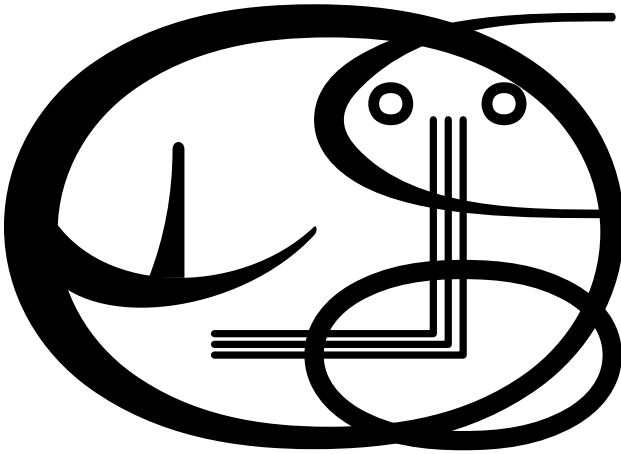


Figure 1: Cat

4.1.1 METAFONT program

The file is 70 lines. Consult anthology.mf.

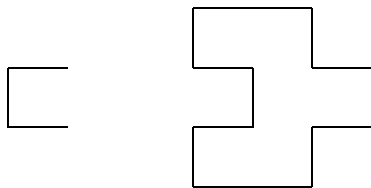
4.2 Hilbert curve

In Wirth's 'Algorithms + Data structures = Programs' the drawing of Hilbert and Sierpiński curves have been treated as essentially recursive. Via the path data structure and copying and rotating of pictures built so far, it can be nicely derecursified.

A Hilbert curve consists of 4 (rotated) copies of a base element connected by 3 straight lines, the 3 edges of a square.

H_0 is a dot. The base element of H_k —a Hilbert curve of order k —is H_{k-1} , $k = 1, 2, \dots$

Below H_1 and H_2 have been drawn.¹⁵



The above picture has been obtained in the spirit of graphics via \TeX alone as follows.

```
$$\unitlength5ex
\hbox{\qquad}
\vbox to3\unitlength{\offinterlineskip
```

```
\vss\W1\S1\E1\vss\vss}
\kern25ex
\vbox to3\unitlength{\offinterlineskip
\S1\W1\N1%rotated H_1
\W1 %connector
\W1\S1\E1%H_1
\S1 %connector
\W1\S1\E1%H_1
\E1 %connector
\N1\E1\S1%rotated H_1
\vss}
}$$
```

4.2.1 METAFONT program

```
message "Hilbert curve, variant; %
the path is built up and drawn at 'one stroke';
%METAFONT experiments, code builds upon
%Wirth's A+DS=P, ppl30-133 (more concise,
% uses path data structure; no redoing
% of already constructed paths.)
%The code has been adapted to build up
%the *path*, and is non-recursive.
%December 1995, cgl@rc.service.rug.nl.
%Kees van der Laan, Hunzeweg 57, 9893PB,
% Garnwerd, Holland
tracingstats:=1;
proofing:=1;screenstrokes;autorounding:=0;
pickup pencircle scaled 0.2pt;
def openit = openwindow currentwindow
from origin to (screen_rows,screen_cols)
at (-40s,15s)enddef;
%
path p; s:=10;
sz:=0;p:=origin;%H_0 size and path
n:=5; %Order of H-curve
for k=1 upto n:%H_1,...H_n consecutively
p:= p transformed (identity rotated 90
reflectedabout (origin,up))--
p shifted ((-sz-1)*s,0)--
p shifted ((-sz-1)*s,(-sz-1)*s)--
p transformed (identity rotated -90
reflectedabout (origin,up)
shifted (-sz*s,(-2sz-1)*s));
sz:=2sz+1;
clearit;draw p; showit;
endfor
end
```

Remarks. Order 4 overflows the 'rounding table' limit (300) with the default autorounding. The connecting straight lines are implicit via --.

4.2.2 POSTSCRIPT program

Joseph Romanovsky has transliterated my (recursive) METAFONT code into the following concise POSTSCRIPT program.

```
/S{0 R rlineto currentpoint stroke moveto}def
/T{90 rotate}def /TM{T 1 -1 scale}def
/H{TM dup
0 gt {1 sub H S TM H S H T S
-1 1 scale H 180 rotate 1 add} if
TM}def
/R 8 def
100 100 moveto
6 H pop
showpage
```

¹⁵Done in \TeX by means of BLUe's format Turtle Graphics.

4.3 Linear construction in space II

Thirty years ago I was caught by Gabo, by his constructive art, especially by his ‘Linear construction in space’-like objects.

From the METAFONT viewpoint this example is all about how to handle a 3D object, that is how to describe, project and draw Gabo’s 3D constructive art.

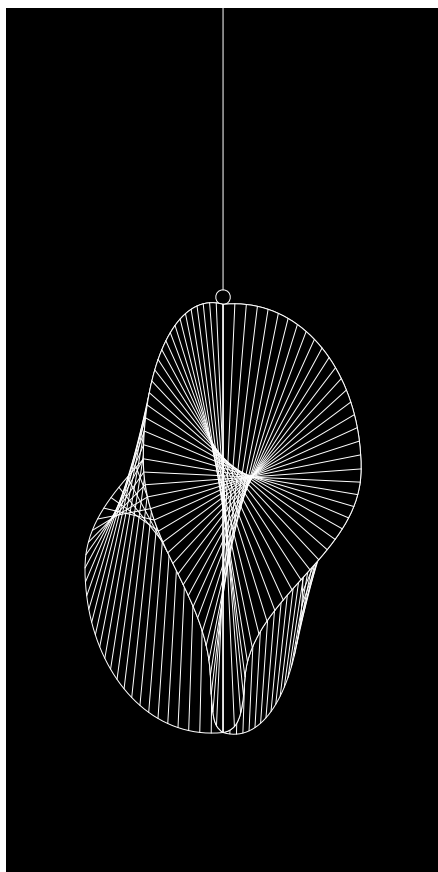


Figure 2: Lin Const in space no.2

How to do this? How to develop a general technique?

4.3.1 Design

It is needed to view the object from different angles. This entailed the use of projection techniques.¹⁶

While programming the object in METAFONT I had to solve

- a how to transform a curve in space?

- b how to preserve shape under projection of (a discretisation of) the curve while joining the projected points by METAFONT’s splines?
- c how to create equidistant points along a curve?
- d how to avoid blurring lines?
- e how to emulate the used (perspex) material?

4.3.2 Coding

I solved b and c by first creating the basic shape of a boundary in 2D, and then replace the curve by a set of (nearly) equidistant points along the curve. The coding of c reads essentially as follows, where p10 takes over from p1.¹⁷

```
p10:=for t:=1 upto 19:
  point .05t of p1..endfor origin;
```

The rotation of a boundary curve, for example from the yz-plane into the xz-plane, is coded simply as follows, where p100 takes over from p10 etc.

```
%in yz-plane (the screen)
p100:= for k=0 upto n-1:
  pointtopair(0,xpart(point k of p10),
             ypart(point k of p10))..
  endfor pointtopair(0,0,0);
%in xz-plane
p200:= for k=0 upto n-1:
  pointtopair(xpart(point k of p10), 0,
             ypart(point k of p10))..
  endfor pointtopair(0,0,0);
```

The projection is done via `pointtopair`, which in its simplest version reads as follows.

```
def pointtopair(expr x,y,z)=
  %Purpose: The projection of a 3D point
  %into a pair in the projection plane.
  %Arguments: x,y,z coordinates of a point
  (-.6x+.8y,-4/13x-3/13y+12/13z)
enddef;
```

Some of the regular surfaces blur the picture.¹⁸ To clear this up I removed the ‘hidden’ line parts. To erase what is hidden is determined by the boundary of what is in front.¹⁹

```
erase fill p100..reverse p200..cycle;
```

The regular surface which is full-blown in sight has been drawn simply via

```
for k=0 step 1 until n:
  draw point k of p100..point n-k of p200; endfor
```

To emulate the ‘light’ caused by the the material perspex I used reverse video as explained in METAFONT Book 115, 118 for the ‘dangerous bend.’

4.3.3 METAFONT program

The file is 120 odd lines. Consult `anthology.mf`.

5 Macro facilities

Macro writing in METAFONT is completely different from macro writing in \TeX . This note is not aimed as a tutorial on macros, it provides a few highlights. An appetizer.

¹⁶See Lauwerier’s ‘Meetkunde met de micro computer,’ for an introduction. It contains many examples in ...BASIC.

¹⁷Note that paths of dynamical length—that is determined at runtime—are created. I fine-tuned this by creating really equidistant points along a curve via the use of METAFONT’s `solve` and Hobby’s `arclength`.

¹⁸In reality Gabo’s aim was that from every viewpoint the object could be seen completely. There are no ‘hidden lines’ in his art. He achieved this by using perspex and nylon.

¹⁹Knuth uses `overdraw`—see for example `ex13.11`—which is very nice. His watchband logo has not been made of perspex apparently.

5.1 FIFO

My favourite FIFO paradigm—first-in-first-out—is implicit in the (var)def parameter handling. For example the macro `max`²⁰ allows as argument a list of undetermined length.

```
max(a)    max(a,b,c)
```

Remark. A variable number of arguments is common in METAFONT, for example `definepixels` and `ilks` can be invoked similarly. This is a consequence of (the abstract) text as parameter ‘type.’ MetaPost’s `buildcycle` macro makes use of this feature too in allowing a list of paths, of undetermined length, as argument. Useful it is.

5.2 Generic macros

For the `max` macro, for example, the type of the arguments can be either numeric, pair or string. This is possible because METAFONT allows for testing for the type of an argument. Powerful this generic—the same macro for all relevant types—feature.

Neat, that abstraction of type and number of arguments, and definitely in agreement with Knuth’s aim²¹

‘The rules are intended to work the way you expect them.’

5.3 Gobbling

Another unusual feature is the infix primary `gobbled`, which not only absorbs the argument *after* but also *before*.²² Infix operators can be defined with primary, secondary or tertiary level of precedence.

5.4 Clipping boundary

Clipping is not provided as such by METAFONT. However, with `cullit` and `cull` a picture can be clipped. Below the current picture is confined to a (scaled) square, and provided with a fret.

```
...%picture so far
%reduce all pixels to 0 or 1
cullit;
%make pixel value of picture 2
%within the square
fill unitsquare scaled 100;
%retain picture within the square
cull currentpicture keeping (2,2);
%draw the boundary
draw unitsquare scaled 100;
```

The clipping by a square boundary is just an example to convey the idea. The approach can be applied to all kinds of shapes, for example to a ring as done by Jackowski in his EuroTeX 96 paper.

5.5 Length of a curve

The macro `length` can be applied to a path with as result the maximum ‘time,’ not the arc length. Hobby applied Simpson’s quadrature rule, which results in the following concise

approximation, because we know the formula of the Bezier spline.²³

$$\int_0^3 \|B'\| dt \approx .5(\|\Delta z_0\| + \|\Delta z_0 + 2\Delta z_1 + \Delta z_2\| + \|\Delta z_2\|)$$

```
vardef lengthpath expr p=save dz; pair dz[];
dz0=point 1 of p - point 0 of p;
dz1=point 2 of p - point 1 of p;
dz2=point 3 of p - point 2 of p;
.5(length(dz0)+length(dz0+2dz1+dz2)+
length(dz2))
enddef;
```

If the path’s name is `p` an invoke might read `lengthpath p`, or for a subpath `lengthpath(subpath (3,6) of p)`.

5.6 Selective loading

In BLUE’s format system the mechanism of selective loading has been used to build a database of tools, pictures and ilks. This functionality can be implemented in METAFONT too. The file to load selectively from consists again of triples: list element tag, a symbolic token, and text enclosed by parentheses and ended by a semicolon. The list element tag macro has 2 arguments: the implicit suffix—which is compared as string with the name of what we want to select—and the text enclosed by parentheses. If the suffix agrees with the required name, a macro of this name is defined with the text as replacement text. For example the definition of the list element tag might read as follows.

```
vardef lst@#(text t)=
if str @#= s:%s=selection key
def @#= t enddef
fi enddef;
input macro.lst
```

The (toy) file `macro.lst` might read as follows

```
lst na (draw unitsquare scaled size);
lst ns (rt);
```

The result with `s:="na"`; reads

```
def na=draw unitsquare scaled size;enddef;
```

6 Pitfall in compatibility

The METAFONT program for my cat processed by MetaPost does not yield the correct result. The moustache has disappeared!?! If we move the moustache code to the end then the correct result is obtained.

What has happened?

In my opinion this is a consequence of that pixels have values, to quote METAFONT Book 109

²⁰Of Appendix D.

²¹Some codes in Appendix D are not that intuitive, however.

²²Peruse Appendix B for these kinds of features.

²³Note that the approximation is only good for sufficient smooth curves. Split up complex curves in simple ones. Another approach is mentioned by Gibbons in his TUG 95 paper. The length of a Bezier spline is bounded by its convex hulls. The smaller the piece the closer the upper and lower bound. Repeated division of the curve and summing the lengths of the pieces yields the length.

Pixels aren't simply 'on' or 'off' when METAFONT is working on a picture; they can be 'doubly on' or 'triple off.' Each pixel contains a small *integer* value, ...

MetaPost's path is apparently just on or off. A `draw <path>` followed by an overlapping fill and unfill makes that the path disappears. It is better to let the `draw <path>` follow the fill and unfill.

Example (*Ring with center*)

The compatible way to program this in METAFONT reads essentially as follows.

```
fill fullcircle scaled 10;
unfill fullcircle scaled 7;
drawdot origin;
```

7 Summarizing my experience

MetaPost combines the best of both worlds: METAFONT's language features are enriched with contours and epsf output.²⁴ Moreover, it allows access to POSTSCRIPT's wealth.

MetaPost can be looked upon as

POSTSCRIPT with a METAFONT user interface

to create pictures to be included in (La) \TeX or troff documents.

Maybe METAFONT/MetaPost will find their niche in history as convenient tools to describe pictures concisely and at a high level.

7.1 What I like of METAFONT

First of all I like very much the quality, stability and its being for free. Next it is available on nearly every platform. Finally, there are the following pleasing details.

- declarative nature of the language
- the meta-ness and the generic aspects
- the generalization of variable, subscripted variable or record field variable into `<tag> <suffix>`
- just 3 kinds of arguments: `expr` (independent of type), `suffix`, and `text`
- the operator definitions with built in priorities (`primarydef` etc.)
- `pen`, `path` and `picture` data structures and operations
- filling and erasing operations
- operations for intersection points
- (nonlinear) interpolation between curves
- various handy 'syntactic sugars,' and
- rich tracing facilities.²⁵

If Descartes' Analytic Geometry is still taught today, it might benefit from METAFONT in visualizing.

²⁴It is nearly upwards compatible. `cullit` and other bitmap operations have to be replaced.

²⁵Agreed, a necessary evil.

²⁶Not to mention Hoenig's typesetting along curved paths.

²⁷Enumerated earlier.

²⁸Bogusław Jackowski and Piotr Pianowski.

7.2 What I missed in METAFONT

In the list of missing items below those denoted by + are provided in MetaPost.

- + outlines or epsf output (it is all about bitmaps of fonts)
- + mixing of text and pictures²⁶
- + general file I/O (writing and reading of pictures)
- + a suitable number range (restricted to the half-open interval $[1/256/256-4096)$)
- + generality of rotating pictures (restricted to a multiple of 90°)
- + dashed/dotted kind of lines
- + arrows
- + clipping
- + shading and greyscales
- + colour
- + to invoke \TeX
- + to make use of POSTSCRIPT facilities
- triple datatype—point in 3D as analogon of `pair`
- tree datastructure (pointer/handle).

Then there is Hobby's 'graph' extension with functionalities²⁷ I did not miss yet, but which I do need for sure when typesetting scientific graphs, gracefully.

7.3 METAFONT/MetaPost as production tool?

History has it that \TeX has been used mainly by scientist with substantial complex copy full of mathematics, tables, or graphics, who wish to publish via the electronic networks, via internet, not in the least in the creative off-off self-publishing world. METAFONT has been used mainly for font production by off-off linguists for non-Latin alphabets.

I don't know of better tools for document production which are so reliable, portable, ubiquitous, open, completely documented, stable, and cooperative towards other tools. With respect to the latter one can think of the various printer and screen drivers, POSTSCRIPT and PDF, and the new hype HTML—HyperText Markup Language. Moreover, the twins \TeX &METAFONT (and descendants) are in the public domain and ported to every platform.

However, one has to learn the systems, to know what is under the hood. \TeX &METAFONT are not of the push-the-button type tools, like washing machines or cars. Therefore, education is paramount.

The METAFONT/POSTSCRIPT experts of the Polish \TeX User Group GUST have reported that the necessary condition for METAFONT/MetaPost to become a production tool is that epsf can function as medium to ease the cooperation with third party tools. As far as I know BoP s.c.²⁸ is the only company with METAFONT in production.

7.4 The future

Maybe we should no longer paste up for figures. What about creating hyperlinks to a picture database? If we want to see the picture we can just click and there it is. This is similar to when we like to hear the music when we read about a composer or read the music notation.

Whatever these new ways will bring, I for one like the complete document on paper.

8 Acknowledgments

Thank you Joseph Romanovsky, Geoffrey Tobin and John Hobby for your friendly support. Piet van Oostrum processed ‘linear construction no. 2’ via MetaPost. The picture came out for sure, at the expense of a large POSTSCRIPT file.

Jos Winnink transcribed the METAFONT pictures into MetaPost. He also lend the usual helping hand in transforming the BLUe scripts into MAPS submissions.

For the details of the references search the file lit.dat which comes with BLUe’s format system.

Have fun, and all the best.

9 Appendix: ToC METAFONT anthology

I hope that the codings mentioned below will contribute to the METAFONT ‘literature.’

```
%Anthology of METAFONT endeavours,
%Started November 1995, last change March 1996
%Version 0, March 1996.
%Author/Compositor: Kees van der Laan,
%Hunzeweg 57, 9893PB, Garnwerd, Holland
% cgl@rc.service.rug.nl.
%This file works under BLUe sky’s METAFONT
%(Public Domain) on the Macintosh.
%(How to scroll through these examples on
% other systems I don’t know.
% Just copy what is of interest for you and
% emulate...
% towards a discipline of METAFONT/MetaPost.)
%For other systems copy the example(s) you
%are interested in build a character
%from it and follow the usual procedures to
%include the font in your (La)TeX,
%or use for example the Jackowski/Rycko
%mftoepsf package to produce epsf.
%For use in MetaPost enclose the elements
%by beginfig ...endfig, I presume.
%
%Disclaimer borrowed from
%Preface METAFONT book:
%‘That’s the way how it is with any
% powerful tool:
% There is always more to learn, and
% there are always better ways to do
% what you have done before.’
%
%Table of Contents
%%%Preliminary, file ant.mac
%-Macros used
% (default) openit
% pointtopair projection
% dash
% turtle movements: east, south, west, north
%-Defaults
%%%ToC proper%%%
```

```
%
%%%Tiles, file: ant.til
%-Roos and variation via interpath
%-Variations via interpath
%-Mondriaan
%-Alhambra tiles (Courtesy M C Escher)
%-Escher tiles
%-Meeting and meeting apart
%-Moriscs ornament (Courtesy J V Romanovsky)
%-Chinese porcelain ornament (Courtesy
% J V Romanovsky)
%-Chinese porcelain ornament II
%-Horak’s tiles
%%%2D figures, file: ant.two
%-Cat (varying pen width, non-linear
% interpolation)
%-Cat II (portable?)
%-Cat III(essential picture)
%-Whirlpool (Courtesy H A Lauwerier)
%-Generalized polygon (sides as flexes)
%-Flexes II (Courtesy B Jackowski)
%-Flexes III(Interpath)
%-2D polygon regular patterns
%-Nails (Courtesy J V Romanovsky)
%-Nails II
%-Nails III
%-Yin-Yang
%-Removing Overlap and Expanded Stroke
%-Turtle graphics: square spiral
%-Spiral of squares (Courtesy H A Lauwerier)
%
%%%3D, file: ant.thr
%-2.5D simplest example: twisted plane
%-Cube and impossible cube (MB 13.7,
% 2.5D variant)
%-Tetraeder
%-Tetraeder and Gabo’s torsions inside
%-Hobby’s pyramid (without text etc)
% in 2.5D
%-Spiral 2.5D
%-Emulating Gabo: doll’s
%-Emulating Gabo: Hyperboloid
%-Emulating Gabo: Linear construction I
%-Emulating Gabo: Linear construction II
%-Emulating Gabo: Spheric theme
%-Emulating Gabo: Spheric theme;
% interpolating surfaces
%-Emulating Gabo: Vertical construction 0
%-Toroid of rotating circles
%-Toroid of rotating circles (2.5D variant)
%-Cube example, varying viewing angles
%-Moebius band, varying viewing angles
%
%%%OP art, file ant.opt
%-Optical illusion (jiggling squares)
%-Optical illusion (parallel?)
%-Op/kinetical art (Courtesy Soto)
%-Knuth meets Vasarely
%-Vasarely II
%-Vasarely III
%-Vasarely IV
%-Vasarely V
%-Vasarely VI
%-Op art pulsing I (Courtesy Jean Larcher)
%-Op art pulsing II(Courtesy Jean Larcher)
%-Op art (Profile)
%
%%%Fractals, file: ant.frc
%-Cantor dust fractal
%-Hilbert curve (Courtsey N Wirth)
%-Sierpinski curve (Courtsey N Wirth)
%-W-curve (Courtsey N Wirth)
%-H-fractal
%-Pythagorean tree
```

```

%-Pythagorean tree, non-recursive
%-Sierpinski square (Courtesy B Jackowski)
%-Sierpinski triangle (recursive
%       with use of save)
%-Sierpinski triangle (without save)
%-Sierpinski carpet (via gambling)
%
%%Various, file ant.var
%-Haralambous' deformations of a 'circle'
%-Calculation of (arc)length of a Bezier
% segment via quadrature
%-Exercise from MB: p.176 (solve use)
%-Exercise from MB: 13.8 (star)
%-Exercise from MB: 13.10 (S-figure)
%-Exercise from MB: 13.10 (S-figure

%       contour via 'expanded stroke')
%-Exercise from MB: 13.11 (Moebius band)
%-Exercise from MB: 13.11 (variant
%       2.5D Moebius band)
%-Example from MB 14 p134 (Interpath use:
%       heart interpolation)
%-Exercise from MB: 15.6 (variant,
%       via paths)
%-Exercise from MB: 20.5 (variant,
%       parameter separator TeXnique?)
%-n faculty (exercise recursion and number range)
%-n asterisks(exercise recursion)
%-quicksort (exercise recursion)
%%end ToC

```

Page layout in L^AT_EX

Piet van Oostrum*
Dept. of Computer Science
Utrecht University

piet@cs.ruu.nl
<http://www.cs.ruu.nl/people/piet>

Abstract

This article describes how to customize the page layout of your LaTeX documents, i.e. how to change page margins and sizes, headers and footers, and the proper placement of figures and tables (collectively called floats) on the page.

Originally this was the documentation of the `fancyheadings` package. It did contain also other info, e.g. advanced use of marks. It has now been upgraded to include more, e.g. the handling of floats. The `fancyheadings` documentation has been upgraded to conform to version 2 of this package¹. For reasons of compatibility with certain operating systems, the name of the package has been changed to `fancyhdr`.

Although this paper uses L^AT_EX_{2 ϵ} commands, most of the techniques can be used with older L^AT_EX versions with appropriate changes.

1 Introduction

A page in a L^AT_EX document is built from various elements as shown in figure 1.

The body contains the main text of the document together with the so called floats (tables and figures).

The pages are constructed by L^AT_EX's output routine, which is quite complicated and should therefore not be modified. Some of the packages described in this paper contains small modifications to the output routine to accomplish things that cannot be done in another way. You should use these packages to get the desired result rather than fiddling with the output routine yourself.

There are a number of things that you must be aware of:

1. The margins on the left are not called `\leftmargin`, but `\evensidemargin` (on even-numbered pages) and `\oddsidemargin` (on odd-numbered pages). In one-sided documents `\oddsidemargin` is used for either. `\leftmargin` is also a valid L^AT_EX parameter but it has a different use (namely the indentation of lists).
2. Most of the parameters should not be changed in the middle of a document. Some changes might work at a pagebreak. If you want to change the height of a single page, you can use the `\enlargethispage` command.

The margin notes area contains small pieces of information created by the `\marginpar` command. On twosided documents the margin notes appear on the left and right alternatively. The margin notes are not on fixed places with respect to the paper but at approximately the same height as the paragraph in which they appear. Due to the algorithm used to decide the placement of margin notes, in a twosided document unfortunately they may appear on the wrong side if they are close to a page break. If you want to put information on fixed places in the margins you may use the technique described in sections 19 and 20.

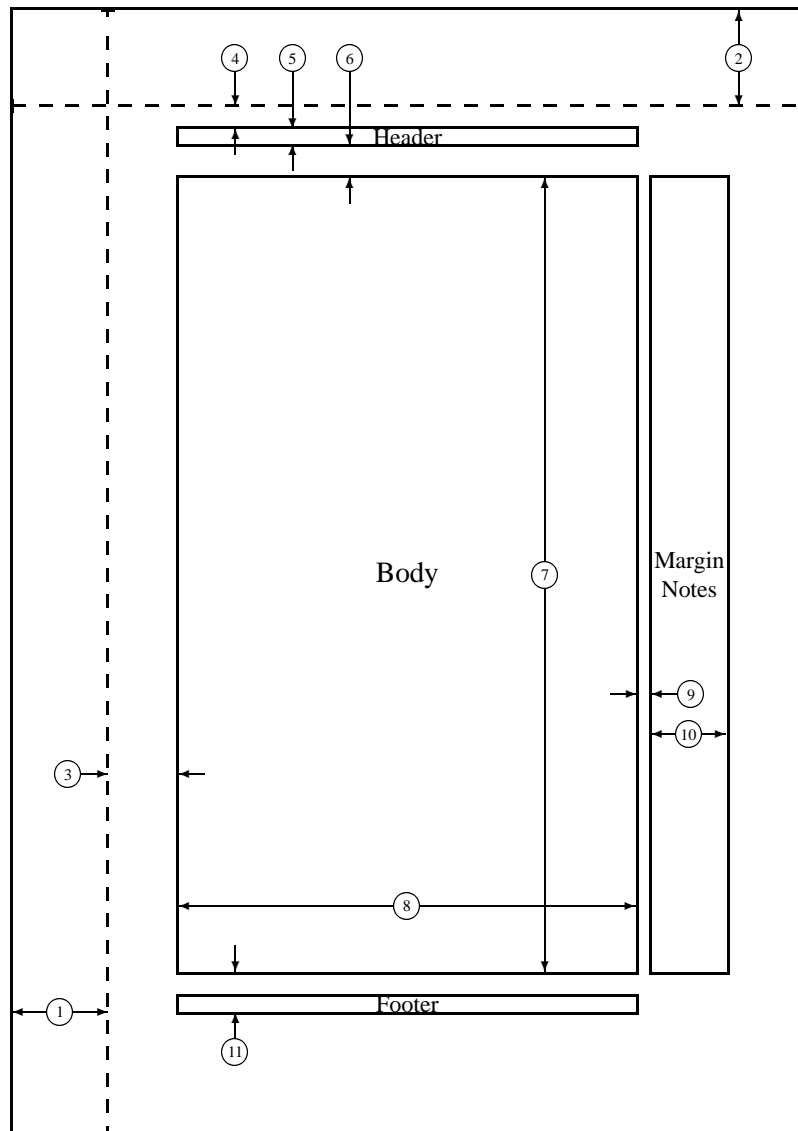
The first part of this paper describes how to change the header and footer areas. The last part describes how to get your floats at the desired place.

2 Page headers and footers

The page headers and footers in L^AT_EX are defined by the `\pagestyle` and `\pagenumbering` commands. `\pagestyle` defines the general contents of the headers and footers (e.g. where the page number will be printed), while `\pagenumbering` defines the format of the page number. L^AT_EX has four standard pagestyles:

*A considerable part of this article was written by George Grätzer (University of Manitoba) in *Notices Amer. Math. Soc.* Thanks, George!

¹this version is due to be released Real Soon Now



- | | | | |
|----|-----------------------|----|----------------------------------|
| 1 | one inch + \hoffset | 2 | one inch + \voffset |
| 3 | \oddsidemargin = 53pt | 4 | \topmargin = 17pt |
| 5 | \headheight = 12pt | 6 | \headsep = 25pt |
| 7 | \textheight = 598pt | 8 | \textwidth = 345pt |
| 9 | \marginparsep = 11pt | 10 | \marginparwidth = 57pt |
| 11 | \footskip = 30pt | | \marginparpush = 5pt (not shown) |
| | \hoffset = 0pt | | \voffset = 0pt |
| | \paperwidth = 597pt | | \paperheight = 845pt |

Figure 1: Page elements. The values shown are those in effect in the current document, not the defaults.

<code>empty</code>	no headers or footers
<code>plain</code>	no header, footer contains page number centered
<code>headings</code>	no footer, header contains name of chapter/section and/or subsection and page number
<code>myheadings</code>	no footer, header contains page number and user supplied information

Although these are useful styles, they are quite limited. Additional page styles can be defined by defining commands of the form `\ps@xxx`. This command is executed when a `\pagestyle{xxx}` is given in the document. The `\ps@xxx` command should define the following commands for the contents of the headers and footers:

<code>\@oddhead</code>	header on odd numbered pages in two-sided documents (on all pages in one-sided)
<code>\@evenhead</code>	header on even numbered pages in two-sided documents
<code>\@oddfoot</code>	footer on odd numbered pages in two-sided documents (on all pages in one-sided)
<code>\@evenfoot</code>	footer on even numbered pages in two-sided documents

These are not user commands, but rather “variables” that are used by L^AT_EX’s output routine. As the command names contain the character ‘@’, they should be defined in a package file, or otherwise be sandwiched between the commands `\makeatletter` and `\makeatother`.

The `\pagenumbering` command defines the layout of the page number. It has a parameter from the following list:

<code>arabic</code>	arabic numerals
<code>roman</code>	lower case roman numerals
<code>Roman</code>	upper case roman numerals
<code>alph</code>	lower case letter
<code>Alph</code>	upper case letter

The `\pagenumbering{xxx}` defines the command `\thepage` to be the expansion of the page number in the given notation `xxx`. The `pagestyle` command then would include `\thepage` in the appropriate place. Additionally the `\pagenumbering` command resets the page number to 1. The `\pagestyle` and `\pagenumbering` apply to the page that is being constructed, so they should be used at a location where it is clear to what page they apply (see section 17).

3 What is fancyhdr

The `fancyhdr` macro package allows you to customize in L^AT_EX your page headers and footers in an easy way. You can define:

- three-part headers and footers
- decorative lines in headers and footers
- headers and footers wider than the width of the text
- multi-line headers and footers
- separate headers and footers for even and odd pages
- different headers and footers for chapter pages
- different headers and footer on pages with floats

Of course, you also have complete control over fonts, uppercase and lowercase displays, etc.

4 Simple use of fancyhdr

To use this package in a L^AT_EX_{2 ϵ} document, place the file `fancyhdr.sty` in a directory/folder where T_EX can find it (normally in the input directory/folder), and include in the preamble of your document after

```
\documentclass{...}
```

the commands²:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

We can visualize the page layout we can create with `fancyhdr` as follows:

²For L^AT_EX_{2.09} you should specify `[fancyhdr]` in the `\documentstyle` instead of the `\usepackage` command.

LeftHeader	CenteredHeader	RightHeader
page body		
LeftFooter	CenteredFooter	RightFooter

The LeftHeader and LeftFooter are left justified; the CenteredHeader and CenteredFooter are centered; the RightHeader and RightFooter are right justified.

We define each of the six “fields” and the two decorative lines separately.

5 A simple example

K. Grant is writing a report to Dean A. Smith, on “The performance of new graduates” with the following page layout:

The performance of new graduates		
page body		
From: K. Grant	To: Dean A. Smith	3

where “3” is the page number. The title: “The performance of new graduates” is bold.

This is accomplished by these commands following `\pagestyle{fancy}`³:

```
\lhead{}
\chead{}
\rhead{\bfseries The performance of new graduates}
\lfoot{From: K. Grant}
\cfoot{To: Dean A. Smith}
\rfoot{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

(The `\thepage` macro displays the current page number. `\bfseries` is the L^AT_EX_{2 ϵ} 's way of selecting bold face.)

This is now fine, except that the first page does not need all these headers and footers. To eliminate all but the centered page number, issue the command

```
\thispagestyle{plain}
```

after the `\begin{document}` and the `\maketitle` commands.

Alternatively, issue

```
\thispagestyle{empty}
```

if you do not want any headers or footers.

In fact the standard L^AT_EX classes have the command `\maketitle` defined in such a way that a `\thispagestyle{plain}` is automatically issued. So if you *do* want the fancy layout on a page containing `\maketitle` you must issue a `\thispagestyle{fancy}` after the `\maketitle`.

6 An example of two-sided printing

Some document classes, such as `book.cls`, print two-sided by default: the even pages and the odd pages have different layouts; other document classes use the `twoside` option to print two-sided.

Now let us print the report two-sided. Let the above page layout be used for the odd (right-side) pages, and the following for the even (left-side) pages:

³Note that version 1 of fancyheadings used the `\setlength` command to change the `\. . . rulewidth` parameters.

E	Even page
O	Odd page
L	Left field
C	Center field
R	Right field
H	Header
F	Footer

Figure 2: Selectors

The performance of new graduates		
page body		
4	From: K. Grant	To: Dean A. Smith

where “4” is the page number.

Here are the commands:

```
\fancyhead{} % clear all fields
\fancyhead[RO,LE]{\bfseries The performance of new graduates}
\fancyfoot[LE,RO]{\thepage}
\fancyfoot[LO,CE]{From: K. Grant}
\fancyfoot[CO,RE]{To: Dean A. Smith}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

We use the more general commands `\fancyhead` and `\fancyfoot`. These have an additional parameter between square brackets that specifies for which pages and/or parts of the header/footer they apply. The first `\fancyhead` command omits this parameter, and thus applies to all header fields. In general this is only useful to get rid of the defaults or a previous definition, as is done here. The selectors that can be used between the square brackets are given in figure 2. Selectors can be combined so `\fancyhead[LE,RO]{text}` will define the field for both the left header on even pages and the right header on odd pages. If you don’t give an E or O the definition applies to both. Similar for LRC. So the use of `\lhead` in the previous section is just an abbreviation for `\fancyhead[L]`. The selectors may be given as uppercase or lowercase letters.

There is also a more general command `\fancyhf` that you can use to combine the specifications for headers and footers. This allows additional selectors H (header) and F (footer). In fact `\fancyhead` and `\fancyfoot` are just `\fancyhf` with H and F prespecified.

Again, you may use `\thispagestyle{plain}` for a simple page layout for page 1.

7 Redefining plain style

Some L^AT_EX commands, like `\chapter`, use the `\thispagestyle` command to automatically switch to the plain page style, thus ignoring the page style currently in effect. To customize even such pages you must redefine the plain page style. As we indicated before you could do this by defining the `\ps@plain` command, but `fancyhdr` gives you an easier way with the `\fancypagestyle` command. This command can be used to redefine existing pagestyles (like plain) or to define new ones, e.g. if part of your document is to use a different pagestyle. This command has two parameters: one is the name of the pagestyle to be defined, the second consists of commands that change the headers and/or footers, i.e. `fancyhead` etc. Also allowed are changes to `\headrulewidth` and `\footrulewidth`. As an example, let us redefine the plain style for the report in Section 6 by making the page number bold.

```
\fancypagestyle{plain}{%
\fancyhf{} % clear all header and footer fields
\fancyfoot[C]{\bfseries \thepage} % except the center
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}}
```

8 The default layout

Let us use the `book.cls` documentclass and the default settings for `fancyhdr`; so we only issue the commands

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

and let `fancyhdr` take care of everything. On the pages where new chapters start, we get a centered page number in the footer; there is no header, and there are no decorative lines.

On an even page, we get the layout:

<i>1.2 EVALUATION</i>	<i>CHAPTER 1. INTRODUCTION</i>
page body	
4	

On an odd page, we get the layout:

<i>CHAPTER 1. INTRODUCTION</i>	<i>1.2 EVALUATION</i>
page body	
3	

where the header text is slanted uppercase.

This default layout is produced by the following commands:

```
\fancyhead[LE,RO]{\slshape \rightmark}
\fancyhead[LO,RE]{\slshape \leftmark}
\fancyfoot[C]{\thepage}
```

The following settings are used for the decorative lines:

```
\headrulewidth 0.4pt
\footrulewidth 0 pt
```

The header text is turned into all uppercase in `book.cls`.

9 The scoop on L^AT_EX's marks

Usually, for documents of class `book` and `report`, you may want to use chapter and section information in the headings (chapter only for one-sided printing), and for documents of class `article`, section and subsection information (section only for one-sided printing). L^AT_EX uses a marker mechanism to remember the chapter and section (section and subsection) information for a page; this is discussed in detail in the *L^AT_EX Companion*, Section 4.3.1.

There are two ways you can use and change the higher- and lower-level sectioning information available to you. The macros: `\leftmark` (higher-level) and `\rightmark` (lower-level) contain the information processed by L^AT_EX, and you can use them directly as shown in Section 8.

The `\leftmark` contains the Left argument of the *Last* `\markboth` on the page, the `\rightmark` contains the Right argument of the *fiRst* `\markboth` or the only argument of the *fiRst* `\markright` on the page. If no marks are present on a page they are “inherited” from the previous page.

You can influence how chapter, section, and subsection information (only two of them!) is displayed by redefining the `\chaptermark`, `\sectionmark`, and `\subsectionmark` commands⁴. You must put the redefinition after the first call of `\pagestyle{fancy}` as this sets up the defaults.

Let us illustrate this with chapter info. It is made up of three parts:

- the number (say, 2), displayed by the macro `\thechapter`
- the name (in English, Chapter), displayed by the macro `\chaptername`

⁴There are similar commands for paragraph and subparagraph but they are seldom used.

Code:	Prints:
<pre>\renewcommand{\chaptermark}[1]{% \markboth{\chaptername \thechapter.\ #1}{}}</pre>	Chapter 2. Do it now
<pre>\renewcommand{\chaptermark}[1]{% \markboth{\MakeUppercase{% \chaptername}\thechapter.% \ #1}{}}</pre>	CHAPTER 2. Do it now
<pre>\renewcommand{\chaptermark}[1]{% \markboth{\MakeUppercase{% \chaptername\thechapter.% \ #1}}{}}</pre>	CHAPTER 2. DO IT NOW
<pre>\renewcommand{\chaptermark}[1]{% \markboth{\ #1}{}}</pre>	Do it now
<pre>\renewcommand{\chaptermark}[1]{% \markboth{\thechapter.\ #1}{}}</pre>	2. Do it now
<pre>\renewcommand{\chaptermark}[1]{% \markboth{\thechapter.% \chaptername.\ #1}{}}</pre>	2. Chapter. Do it now

Figure 3: Marker variants

- the title, contained in the argument of `\chaptermark`.

Figure 3 shows some variants for “Chapter 2. Do it now” (the last example is appropriate in some non-English languages). The % signs at the end of the lines are to prevent unwanted space. Normally you would continue the lines and remove these % signs⁵.

For the lower-level sectioning information, do the same with `\markright`.

So if “Section 2.2. First steps” is the current section, then

```
\renewcommand{\sectionmark}[1]{\markright{\thesection.\ #1}}
```

will give “2.2. First steps”

Redefining the `\chaptermark` and `\sectionmark` commands may not eliminate all uppercaseness. E.g. the bibliography will have a title of BIBLIOGRAPHY in the header, as the `\MakeUppercase` is explicitly given in the definition of `\thebibliography`. Similar for INDEX etc. If you don’t want to redefine these commands, you can use the `\nouppercase` command that `fancyhdr` makes available in the header and footer fields. Note that this may screw other things, like uppercase roman numerals in your headers, so it should be used with care. Essentially this command typesets its argument in an environment where `\MakeUppercase` and `\uppercase` are changed into do-nothing operations.

```
\lhead{\nouppercase{\rightmark}}
\rhead{\nouppercase{\leftmark}}
```

It should be noted that the L^AT_EX marking mechanism works fine with chapters (which always start on a new page) and sections (which are reasonably long). It does not work quite as well with short sections and subsections. This is a problem with L^AT_EX, not with `fancyhdr`.

As an example let’s take a page layout where the leftmarks are generated by the sections and the rightmarks by the subsections (as is default in the `article` class). Take a page with some short sections, e.g.

```
Section 1.
subsection 1.1
```

⁵the `\MakeUppercase` command is used in L^AT_EX_{2_ε} to generate uppercase text, while in L^AT_EX 2.09 `\uppercase` is used. The difference is that `\MakeUppercase` also deals with non-ASCII letters. `Fancyhdr` defines `\MakeUppercase` to be an alias for `\uppercase` if it isn’t defined.

subsection 1.2
Section 2.

As the leftmark contains the *last* mark of the page it will be “Section 2.”, and the rightmark will be “subsection 1.1” as it will be the *first* mark of the page. So the page header info will combine section 2 with subsection 1.1 which isn’t very nice. The best you can do in these cases is use only the `\rightmarks` and redefine `\sectionmark` accordingly. A L^AT_EX command `\firstleftmark` would also be a nice addition (see the `extramarks` package in section 18).

Another problem with the marks in the standard L^AT_EX classes is that the higher level sectioning commands (e.g. `\chapter`) call `\markboth` with an empty right argument. This means that on the first page of a chapter (or a section in article style) the `\rightmark` will be empty. If this is a problem you must manually insert extra `\markright` commands or redefine the `\chaptermark` (`\sectionmark`) commands to issue a `\markboth` command with two decent parameters.

As a final remark you should also note that the `*` forms of the `\chapter` etc. commands do *not* call the mark commands. So if you want your preface to set the header info but not be numbered nor be put in the table of contents, you must issue the `\markboth` command yourself, e.g.

```
\chapter*{Preface\markboth{Preface}{}}
```

Entering the `\markboth` command inside the `\chapter*` insures that the mark will not be separated from the title by a page break. Of course with `\chapter*` this wouldn’t be a problem if you put the mark command after the chapter title, as the `\chapter*` command starts at a new page. However with a `\section*` it could be dangerous to say:

```
\section*{Preface}
\markboth{Preface}{}
```

as a page break may occur between the two commands.

10 Dictionary style headers

Dictionaries and concordances usually have a header containing the first word defined on the page or both the first and the last words. This can easily be accomplished with `fancyhdr` and L^AT_EX’s mark mechanism. Of course if you use the marks for dictionary style headers, you cannot use them for chapter and section information, so if there are also chapters and sections present, you must redefine the `\chaptermark` and `\sectionmark` to make them harmless:

```
\renewcommand{\chaptermark}[1]{}
\renewcommand{\sectionmark}[1]{}

```

Now you do a `\markboth{#1}{#1}` for each dictionary or concordance entry `#1` and use `\rightmark` for the first entry defined on the page and `\leftmark` for the last one.

If you want to use a header entry of the form `firstword–lastword` it would be nice if this would be reduced to just the form `firstword` if both are the same. This could happen if there is just one entry on the page. In this case a test must be made to check if the marks are the same. However, T_EX’s marks are strange beasts, which cannot be compared out of the box with the plain T_EX `\if` commands. Fortunately the `ifthen` package works well:

```
\newcommand{\mymarks}{
  \ifthenelse{\equal{\leftmark}{\rightmark}}
    {\rightmark} % if equal
    {\rightmark--\leftmark} % if not equal
}
\fancyhead[LE,RO]{\mymarks}
\fancyhead[LO,RE]{\thepage}

```

Dictionaries are often done with two columns. Unfortunately there is a bug in L^AT_EX’s `twocolumn` option which causes some marks to be lost. If you use the package in figure 4, this will be solved⁶⁷.

11 Fancy layouts

You can make a multi-line field with the `\` command. It is also possible to put extra space in a field with the `\vspace` command. Note that if you do this you will probably have to increase the height of the header (`\headheight`) and/or of the footer (`\footskip`), otherwise you may get error messages “Overfull \vbox ...has occurred while \output is active”⁸. See Section 4.1 of the L^AT_EX *Companion* for detail.

For instance, the following code will place the section title and the subsection title of an article in two lines in the upper right hand corner:

⁶This is a simplification of the actual code. Get the real package from the CTAN sites.

⁷The `multicol` package uses a similar technique.

⁸If you use 11pt or 12pt you will probably also have to do this, because L^AT_EX’s defaults are quite small

```

% fixmarks.sty:
% Patch LaTeX's output routine to handle marks correctly with two columns.
% Joe Pallas <pallas@edu.stanford.neon>
% Corrected by Piet van Oostrum <piet@cs.ruu.nl> on Feb 5, 1993, Oct 5, 1994

\def\@outputdblcol{\if@firstcolumn \global\@firstcolumnfalse
% Remember the marks from the first column
  \global\setbox\@leftcolumn\copy\@outputbox
  \splitmaxdepth=\maxdimen \cbadness=10000
  \setbox\@outputbox\vsplit\@outputbox to\maxdimen
  \xdef\@firstcoltopmark{\topmark}%
  \xdef\@firstcolfirstmark{\splitfirstmark}%
  \ifx\@firstcolfirstmark\empty\global\let\@setmarks\relax\else
  \gdef\@setmarks{\let\firstmark\@firstcolfirstmark
    \let\topmark\@firstcoltopmark}%
  \fi
% End of change
\else \global\@firstcolumntrue
  \setbox\@outputbox\vbox{\hbox to\textwidth{\hbox to\columnwidth
    {\box\@leftcolumn \hss}\hfil \vrule width\columnseprule\hfil
    \hbox to\columnwidth{\box\@outputbox \hss}}}\@combinedblfloats
% Override current first and top with those of first column if necessary
  \@setmarks
% End of change
  \@outputpage \beginngroup \@dblfloatplacement \@startdblcolumn
  \@whilsw\if@fcolmade \fi{\@outputpage\@startdblcolumn}\endgroup
\fi}

```

Figure 4: Fix marks in twocolumn style

```

\documentclass{article}
\usepackage{fancyhdr}
\pagestyle{fancy}
\addtolength{\headheight}{\baselineskip}
\renewcommand{\sectionmark}[1]{\markboth{#1}{}}
\renewcommand{\subsectionmark}[1]{\markright{#1}}
\rhead{\leftmark\rightmark}

```

You can also customize the decorative lines. You can make the decorative line in the header quite thick with

```
\renewcommand{\headrulewidth}{0.6pt}
```

or you can make the decorative line in the footer disappear with

```
\renewcommand{\footrulewidth}{0pt}
```

The decorative lines, themselves, are defined in the two macros `\headrule` and `\footrule`. For instance, if you want a dotted line rather than a solid line in the header, redefine the command `\headrule`:

```
\renewcommand{\headrule}{\vbox to 0pt{\hbox
to\headwidth{\dotfill}\vss}}
```

There is one additional parameter that you can set: `\footruleskip`. It defines the distance between the decorative line in the footer and the top of the footer text line. By default it is set to 30% of the normal line distance. You may want to adjust it if you use unusually large or small fonts in the footer. Change it with `\renewcommand`.

12 Two book examples

The following definitions give an approximation of the style used in L. Lamport's L^AT_EX book.

Lamport's header overhangs the outside margin. This is done as follows.

The width of headers and footers is `\headwidth`, which by default equals the width of the text: `\textwidth`. You can make the width wider (or narrower) by redefining `\headwidth` with the `\setlength` and `\addtolength` commands. To overhang the outside margin where the marginal notes are printed, add both `\marginparsep` and `\marginparwidth` to `\headwidth` with the commands:

```
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
```

You must issue these commands *after* the first `\pagestyle{fancy}` command as this will establish the default for `\headwidth`⁹.

⁹In version 2 of `fancyhdr` it is usually safe to do it before, but it is still advised to do after.

And now a complete definition of Lamport's book style:

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
\addtolength{\headwidth}{\marginparsep}
\addtolength{\headwidth}{\marginparwidth}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\fancypagestyle{plain}{%
  \fancyhead{} % get rid of headers
  \renewcommand{\headrulewidth}{0pt} % and the line
}
```

Notice that the `\chaptermark` and `\sectionmark` commands have been redefined to eliminate the chapter numbers and the uppercasesness.

For the second example, we take the $\overline{\text{L}}\text{A}\overline{\text{T}}\text{E}\overline{\text{X}}$ book.

Chapter pages have no headers or footers. So we declare

```
\thispagestyle{empty}
```

for every chapter page, and we do not need to redefine plain.

Chapter and section titles appear in the form: 2. DO IT NOW, so we have to redefine `\chaptermark` and `\sectionmark` as follows (see Section 9):

```
\renewcommand{\chaptermark}[1]%
  {\markboth{\MakeUppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
  {\markright{\MakeUppercase{\thesection.\ #1}}}
```

In an even-header, the page number is printed as the LeftHeader and the chapter info as the RightHeader; in an odd-header, the section info is printed as the LeftHeader and the page number as the RightHeader. The CenteredHeaders are empty. There are no footers.

There is a decorative line in the header. It is 0.5pt wide, so we need the commands:

```
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
```

The font used in the headers is 9 pt bold Helvetica. The PSNFSS system by Sebastian Rahtz uses the short (Karl Berry) name `phv` for Helvetica, so this font is selected with the commands:

```
\fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont
```

(See Sections 7.6.1 and 11.9.1 of the *L^AT_EX Companion*.) Let us define a shorthand for this:

```
\newcommand{\helv}{%
  \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
```

Now we are ready for the page layout:

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
\renewcommand{\chaptermark}[1]%
  {\markboth{\MakeUppercase{\thechapter.\ #1}}{}}
\renewcommand{\sectionmark}[1]%
  {\markright{\MakeUppercase{\thesection.\ #1}}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\newcommand{\helv}{%
  \fontfamily{phv}\fontseries{b}\fontsize{9}{11}\selectfont}
\fancyhf{}
\fancyhead[LE,RO]{\helv \thepage}
\fancyhead[LO]{\helv \rightmark}
\fancyhead[RE]{\helv \leftmark}
```

13 Special page layout for float pages

Some people want to have a special layout for float pages (pages only containing floats). As these pages are generated autonomically by L^AT_EX, the user doesn't have any control over them. There is no `\thispagestyle` for float pages and any change of the page style will at least also affect the page before the float page. With `fancyhdr`, however, you can specify in each of the header- or footer fields

```
\iffloatpage{value for float page}{value for other pages}
```

You can even use this to get rid of the decorative line on float pages only by defining:

```
\renewcommand{\headrulewidth}{\iffloatpage{0pt}{0.4pt}}
```

Sometimes you may want to change the layout also for pages that contain a float on the top of the page or a float on the bottom of the page.

`fancyhdr` gives you the commands `\iftopfloat` and `\ifbotfloat` similar to `\iffloatpage`.

Note: Marks in floats will not be visible in L^AT_EX's output routine, so it is not useful to put marks in floats. So there is currently no way to let a float (e.g. a figure caption) influence the page header or footer.

14 Those blank pages

In the `book` class when the `openany` option is not given or in the `report` class when the `openright` option is given, chapters start at odd-numbered pages, half of the time causing a blank page to be inserted. Some people prefer this page to be completely empty, i.e. without headers and footers. This cannot be done with `\thispagestyle` as this command would have to be issued on the *previous* page. There is, however, no magic necessary to get this done:

```
\clearpage{\pagestyle{empty}\cleardoublepage}
```

As the `\pagestyle{empty}` is enclosed in a group it only affects the page that may be generated by the `\cleardoublepage`. You can of course put the above in a private command. If you want to have this done automatically at each chapter start or when you want some other text on the page then you must redefine the `\cleardoublepage` command.

```
\makeatletter
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
  \hbox{}
  \vspace*{\fill}
  \begin{center}
    This page intentionally contains only this sentence.
  \end{center}
  \vspace{\fill}
  \thispagestyle{empty}
  \newpage
  \if@twocolumn\hbox{} \newpage\fi\fi\fi}
\makeatother
```

15 N of M style page numbers

Some document writers prefer the pages to be numbered as *n* of *m* where *m* is the number of pages in the document. There is a package `nofm.sty` available, but some versions of it are defective, and most don't work with `fancyhdr` because they take over the complete page layout. For L^AT_EX_{2 ϵ} there is a package `lastpage` available which you can use with `fancyhdr` as follows:

```
\usepackage{lastpage}
...
\cfoot{\thepage\ of \pageref{LastPage}}
```

If you are still using L^AT_EX_{2.09} and you are not able to switch to L^AT_EX_{2 ϵ} you can use the L^AT_EX_{2.09} compatible `lastpage209.sty` which is defined as follows:

```
\let\origenddocument=\enddocument
\def\enddocument{\clearpage\if@filesw
  {\addtocounter{page}{-1} \immediate\write\@mainaux
   {\string\newlabel{LastPage}{{}{\thepage}}}}\origenddocument}
```

The value of the `LastPage` label can be used to make different headers or footers on the last page of a document. E.g. if you want the footer of every odd page, except if it is the last one, to contain the text "please turn over", this can be done as follows¹⁰:

```
\usepackage{lastpage}
\usepackage{ifthen}
...
\rfoot{\ifthenelse{\isodd{\value{page}} \and \not
  \value{page}=\pageref{LastPage}{please turn over}}}
```

¹⁰This requires a reasonably recent version of the `ifthen` package.

16 Chapter or section related page numbers

In technical documentation very often page numbers are used of the form 2-10 where the first number is the chapter number and the second is the pagenummer relative to the chapter. Sometimes section is used rather than chapter. The package `chappg` can be used to get this format. If you want to change the layout e.g. use a dot rather than a dash, or a section rather than a chapter, you should make a private copy and edit it.

This package redefines `\thepage` as `\arabic{chapter}-\arabic{page}`. Unfortunately this gives numbers rather than letters for appendices. A better definition would be `\thechapter-\arabic{page}` but you can give this definition yourself after the `\usepackage{chappg}` command. What the package also does is reset the page number to 1 at the beginning of each chapter.

There is a fundamental difference between the page numbering of the style “*m of n*” as described in the previous section and the current one. The *m of n* style is only used in the page header or footer, but not in the table of contents, index, or references like “*See page xx*”. Therefore it does not change the command `\thepage`. The page numbering style “2-10”, however should be used in all references to the page number, therefore it must be done by redefining `\thepage`.

17 When to change the headers and footers?

Sometimes you want to change the header or footer layout in the course of a document. Some of these changes can be accomplished by using the mark mechanism as may be seen in section 9 and 18. However, sometimes we want a more drastic change, e.g. to change the page numbering from roman to arabic (with `\pagenumbering`), to change one of the `fancyhdr` fields or to change to another page style. Sometimes you may be surprised to find the change to occur too early. In general the above mentioned changes take effect immediately, i.e. on the page that is currently being built. If you want the change to take effect at the next page you must make sure that the current page is finished. In most cases this can be done by issuing a `\clearpage` command before any of the above mentioned changes. If this is not possible you can use the `afterpage` package with:

```
\afterpage{\thead{new value}} or \afterpage{\pagenumbering{roman}. You cannot use
\afterpage to change the \pagestyle as the commands issued by \afterpage are local in a group, and the
\pagestyle command makes only local changes. The \pagenumbering and the fancyhdr commands make global
changes so they will work, as will the \thispagestyle command.
```

It should be noted that although the `fancyhdr` commands like `\fancyhead` take effect immediately, this does not mean that any “variables” used in these commands get the value they have at the place where these commands are given. E.g. if `\fancyfoot[C]{\thepage}` is given the page number that will be inserted in the footer is not the page number of the page where this command is given, but rather the page number of the actual page where the footer is constructed. Of course for the page number this is what you expect, but it is also true for other commands.

So if you have a book where each chapter is written by a different author and you want the name of the author in the lower left-hand corner you can use the following commands:

```
\newcommand{\TheAuthor}{}
\newcommand{\Author}[1]{\renewcommand{\TheAuthor}{#1}}
\lfoot{\TheAuthor}
```

and start each chapter with the command `\Author{Real Name}`. If however, the author name would be changed before a page is completed the wrong author could come in the footer. This would be the case if you gave the above command *before* the `\chapter` command rather than after it. Another source of problems is the fact that T_EX’s output routine processes commands ahead, so it may already have processed some commands that produce text that will appear on the next page. See the next section for an example.

18 Headers and footers induced by the text

We have seen how we can use L^AT_EX’s marks to get information from the document contents to the headers and footers. The marks mechanism is the only reliable mechanism that you can use to get changing information to the headers or footers. This is because L^AT_EX may be processing your document ahead before deciding to break the page.

Sometimes the two marks that L^AT_EX offers are not enough. An example is the following:

If a solution to an exercise goes across a page break, then I would like to have “(Continued on next page...)” at the bottom of the first page and “(Continued...)” at the top in the margin of the next page.

You cannot use L^AT_EX’s mark mechanisms for this if you also want to use chapter and section information.

The code from figure 5 constitutes a package that gives you two extra marks that can be used in this situation¹¹. Here is a way to use this package:

¹¹After I made this package I discovered a package `secret.sty` that does a similar thing to mark confidential paragraphs if they cross a page boundary. It does it, however, by changing the output routine.

```

% extramarks.sty
\def\@leftmark#1#2#3#4{#1}
\def\@rightmark#1#2#3#4{#2}

\def\markboth#1#2{{\def\protect{\noexpand\protect\noexpand}
\let\label\relax \let\index\relax \let\glossary\relax
\expandafter\@markboth\@themark{#1}{#2}
\mark{\@themark}}\if@nbreak\ifvmode\nobreak\fi\fi}
\def\markright#1{{\def\protect{\noexpand\protect\noexpand}
\let\label\relax \let\index\relax \let\glossary\relax
\expandafter\@markright\@themark
{#1}\mark{\@themark}}\if@nbreak\ifvmode\nobreak\fi\fi}
\def\@markright#1#2#3#4#5{\gdef\@themark{#1}{#5}{#3}{#4}}
\def\@markboth#1#2#3#4#5#6{\gdef\@themark{#5}{#6}{#3}{#4}}
\def\leftmark{\expandafter\@leftmark\botmark{}}{}{}{}
\def\rightmark{\expandafter\@rightmark\firstmark{}}{}{}{}
\def\firstleftmark{\expandafter\@leftmark\firstmark{}}{}{}{}
\def\lastrightmark{\expandafter\@rightmark\botmark{}}{}{}{}

\def\@themark{}}{}{}{}

\def\extramarks#1#2{{\def\protect{\noexpand\protect\noexpand}
\let\label\relax \let\index\relax \let\glossary\relax
\expandafter\@markextra\@themark{#1}{#2}
\mark{\@themark}}\if@nbreak\ifvmode\nobreak\fi\fi}
\def\@markextra#1#2#3#4#5#6{\gdef\@themark{#1}{#2}{#5}{#6}}
\def\firstxmark{\expandafter\@firstxmark\firstmark{}}{}{}{}
\def\topxmark{\expandafter\@firstxmark\topmark{}}{}{}{}
\def\lastxmark{\expandafter\@lastxmark\botmark{}}{}{}{}
\def\@firstxmark#1#2#3#4{#3}
\def\@lastxmark#1#2#3#4{#4}

```

Figure 5: Package for extra marks in L^AT_EX

```

\usepackage{extramarks}
...
\pagestyle{fancy}
\lhead{\firstxmark}
\rfoot{\lastxmark}
...
\extramarks{}{Continued on next page\ldots}
Some text that may or may not cross a page boundary...
\extramarks{Continued\ldots}{}

```

Note that the `\extramarks` command must be close to the text, i.e. no empty lines (paragraph boundaries) should intervene. Otherwise the page may be broken at that boundary and the extramarks would come on the wrong page.

There are two new marks that can be used in the page layout with this package: If commands of the form `\extramarks{ m_1 }{ m_2 }` are given `\firstxmark` gives you the first m_1 value and `\lastxmark` gives you the last m_2 value of the current page. It also gives you the `\firstleftmark` and `\lastrightmark` commands that complement the standard L^AT_EX marks.

To stress the point that marks are the correct way to do this, let me give you a “solution” that will not work¹²:

```

\lhead{Continued}
\rfoot{Continued on next page\ldots}
Some text that may or may not cross a page boundary...
\lhead{}
\rfoot{}

```

You may be tempted to think that the first `\lhead` and `\rfoot` will be in effect when T_EX breaks the page in the middle of the text, and the last ones when the page breaks after the text. This is not true as the whole paragraph (including the last definitions) will be processed before T_EX considers the page break, so at the time of the page break the last definitions are effective, whether the page break occurs inside the text or outside of it. Putting a paragraph boundary between the text and the last definitions will not work either, because you don’t want the first definitions to be in effect when T_EX decides to break the page exactly at this boundary. Actually the marks mechanism was invented to get rid of these kinds of problems.

In the above example the text “Continued” appears in the page header. It may be nicer to put it in the margin. This can be easily accomplished by positioning it at a fixed place relative to the page header. In plain T_EX you would use a concoction

¹²Actually there is another way but it requires two L^AT_EX passes: you can put `\label` commands before and after the text and compare the `\pagerefs`.

of `\hbox` to `0pt`, `\vbox` to `0pt`, `\hskip`, `\vskip`, `\hss` and `\vss` but fortunately L^AT_EX's `picture` environment gives a much cleaner way to do this. In order not to disturb the normal header layout we put the text in a zero-sized `picture`. Generally this is the best way to position things on fixed places on the page. You can then also use the normal headings. See also section 20 for another example of this technique.

```
\thead{\setlength{\unitlength}{\baselineskip}%
\begin{picture}(0,0)
  \put(-2,-3){\makebox(0,0)[r]{\firstxmark}}
\end{picture}\leftmark}
```

This solution can of course also be used for the footer. Make sure you put the `picture` as the first thing in left-hand side entries and last in right-hand side ones.

Finally you may want to put “(Continued...)” in the *text* rather than in the header or the margin. Then you have to use the `afterpage` package. We also decide to make a separate environment for it.

```
\newenvironment{continued}{\par
  \extramarks{}{Continued on next page\ldots}
  \afterpage{\noindent\firstxmark\vspace{1ex}}
}{\extramarks{(Continued\ldots)}{\par}}
```

It is a bit dangerous to use `\firstxmark` outside the page layout routine, but apparently with `\afterpage` this works. If you would need the information further on in the page you must remember the state of the marks in your own variable. You can set this in one of the `fancyhdr` fields. For example if you want to add something *after* the broken piece of text you can use the following:

```
\newcommand{\mysaved}{}

\newenvironment{continued}{\par
  \extramarks{}{Continued on next page\ldots}
}{\extramarks{(Continued\ldots)}{\par\vspace{1ex}\mysaved}}
\thead{\leftmark}
\chead{\ifthenelse{\equal{\lastxmark}{} }
  {\gdef\mysaved{}}
  {\gdef\mysaved{\noindent[Continued from previous page]}}}
```

If you want to include one of the marks or other varying information in the saved text, you must use `\xdef` rather than `\gdef`.

19 A movie

If you put at each page on the same place a picture that slightly changes from page to page you can get a movie-like effect by flipping through the pages. You can create such a movie easily with `fancyhdr`. For simplicity we assume that each picture is in a `postscript` (EPS) file called `pic(n).ps` where $\langle n \rangle$ is the page number and that we use the `graphics` or `graphicx` package¹³. To put the movie in the righthand side bottom corner the following will work:

```
\rfoot{\setlength{\unitlength}{1mm}
\begin{picture}(0,0)
  \put(5,0){\includegraphics{pic\thepage.ps}}
\end{picture}}
```

Notice that the `\unitlength` parameter should be set locally in the `fancyhdr` field in order to avoid unwanted interference with its value in the text.

20 Thumb-indexes

Some railroad guides and expensive bibles have so called *thumb-indexes*, i.e. there are marks on the sides of the pages that indicate where the chapters are. You can create these by printing black blobs in the margin of the pages. The vertical position should be determined by the chapter number or some other counter. As the position is independent of the contents of the page, we print these blobs as part of the header in a zero-sized `picture` as described in the previous section.

Of course we have to take care of two-sided printing, and we may want to have an index page with all the blobs in the correct position. The solution requires some hand-tuning to get the blobs nicely spaced out vertically. For the application that I had there were 12 sections, so I made the blobs 18 mm apart, i.e. 9 mm blob separated by 9 mm whitespace. In order to avoid calculations they are set in a `picture` environment with the `\unitlength` set to 18 mm. Page numbers are set in the headers at the outer sides, and the blobs are attached to these. In this example the section numbers are used to position the blobs, but you can replace this with any numeric value. See figure 6 for the resulting overview page and figure 7 for the code.

¹³If you use an older version of L^AT_EX you could use the `epsf` or `epsfig` package.

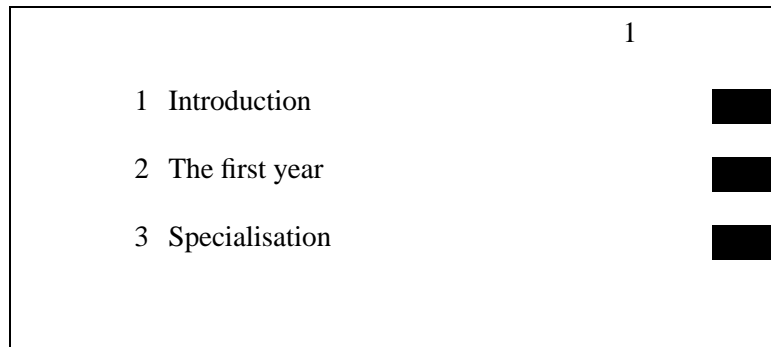


Figure 6: Thumb-index overview page

```

\setlength{\unitlength}{18mm}
\newcommand{\blob}{\rule[-.2\unitlength]{2\unitlength}{.5\unitlength}}

\newcommand\rblob{\thepage
  \begin{picture}(0,0)
    \put(1,-\value{section}){\blob}
  \end{picture}}

\newcommand\lblob{%
  \begin{picture}(0,0)
    \put(-3,-\value{section}){\blob}
  \end{picture}%
  \thepage}

\pagestyle{fancy}
\cfoot{}

\newcounter{line}
\newcommand{\secname}[1]{\addtocounter{line}{1}%
  \put(1,-\value{line}){\blob}
  \put(-7.5,-\value{line}){\Large \arabic{line}}
  \put(-7,-\value{line}){\Large #1}}

\newcommand{\overview}{\thepage
  \begin{picture}(0,0)
    \secname{Introduction}
    \secname{The first year}
    \secname{Specialisation}
    ...etc...
  \end{picture}}

\begin{document}
\fancyhead[R]{\overview}\mbox{}\newpage % This produces the overview page
\fancyhead[R]{} % Front matter may follow here
\clearpage
\fancyhead[RE]{\rightmark}
\fancyhead[RO]{\rblob}
\fancyhead[LE]{\lblob}
\fancyhead[LO]{\leftmark}
...

```

Figure 7: Thumb-index code

21 Float placement

Floats are page elements that float with respect to the rest of the document. Standard floats are tables and figures, but with the `float` package you can easily make new ones, like algorithms. Most of the time floats work satisfactory, but sometimes L^AT_EX seems too stubborn to do what you want. This section describes how you can influence L^AT_EX so that it will do most of the time what you want. There might, however be some pathological cases where it is impossible to convince L^AT_EX to do things your way. In the following we will use figures as an example but everything applies to other floats as well.

The most encountered problems with floats are:

1. You want a float at a certain position in the text, but L^AT_EX moves it, usually to the next page.
2. From a certain point, L^AT_EX moves all your floats to the end of the document or the end of a chapter.
3. L^AT_EX complains about “Too many floats”.

In the first two cases you must first check if you have given the correct “placement” parameter to you float, e.g. `\begin{figure}[htp]` specifies that your figure may be placed either: Here (i.e. in the text position where the command is given), on the Top of a page (which may be the page where you put the command), or on a separate Page of floats. You could also have specified “b” for Bottom of the page. The order of the letters is insignificant, you cannot force L^AT_EX to try Bottom first and then Top by specifying `[bt]`.

If L^AT_EX doesn’t put the float at the place where you expected it, it is usually caused by the following:

1. The float didn’t fit on the page. In this case it has to move to the next page or even further. If you didn’t specify either `[t]` or `[b]` in the position parameter, L^AT_EX must save it until it has enough for a page of floats. So don’t specify only `[h]`. If you want to give L^AT_EX a chance to put the float on a page of floats, you must also specify “p”.
2. The placement would violate the constraints imposed by L^AT_EX’s float placement parameters. This is one of the most occurring causes and it can easily be corrected by changing the parameters. Here is a list of them:

Counters – change with <code>\setcounter</code>		
<code>topnumber</code>	max. number of floats at top of page	2
<code>bottomnumber</code>	max. number of floats at bottom of page	1
<code>totalnumber</code>	max. number of floats on a page	3
Other – change with <code>\renewcommand</code>		
<code>\topfraction</code>	max fraction of page for floats at top	0.7
<code>\bottomfraction</code>	max fraction of page for floats at bottom	0.3
<code>\textfraction</code>	min fraction of page for text	0.2
<code>\floatpagefraction</code>	min fraction of floatpage that should have floats	0.5

There are also some others for double column floats in two-column documents.

The values in the righthand column are the defaults for the standard L^AT_EX classes. Other classes could use different defaults. As you see with the default values a float will not be put in the bottom of a page if its height is more than 30% of the page height. So if you specify `[hb]` for a float which is taller it has to move to a float page. But if it is less than 50% of the page height it will have to wait until some more floats are given before a float page can be filled to satisfy the `\floatpagefraction` parameter. If you have this kind of behaviour you can easily adapt the parameters, e.g. with:

```
\renewcommand{\textfraction}{0.05}
\renewcommand{\topfraction}{0.95}
\renewcommand{\bottomfraction}{0.95}
\renewcommand{\floatpagefraction}{0.35}
\setcounter{totalnumber}{5}
```

You may want to be careful not to make `\floatpagefraction` too small, otherwise you may get too many small floatpages.

You can force L^AT_EX to ignore most of the parameters for one specific float occurrence by including an exclamation mark (!) in the placement parameters, e.g.

```
\begin{figure}[!htb]
```

Floats which contain a “t” in the position parameter could be placed before the place where they are referenced (but on the same page). This is normal behaviour for L^AT_EX but some people just don’t like it. There are a number of ways to prevent this:

1. Of course deleting the “t” will help, but in general this is undesirable, as you may want the float to be placed at the top of the next page.
2. use the `flafter` package which causes floats never to be placed “backwards”.

3. use the command `\suppressfloats[t]`¹⁴. This command will cause floats for the top position *on this page* to be moved to the next page. This can also be done with `[b]` or without parameter for all floats on this page.

If in spite of all your attempts L^AT_EX still moves your floats to the end of the document or the end of a chapter, you can insert a `\clearpage` command. This will start a new page and insert all pending floats before continuing. If it is undesirable to have a pagebreak you can use the `afterpage` package and the following command:

```
\afterpage{clearpage}
```

This will wait until the current page is finished and then flush all outstanding floats. In some pathological circumstances `afterpage` may give strange results, however.

Finally, if you want a float only at the place where you define it, without L^AT_EX moving it whatsoever, you can use the `float` package and give the command:

```
\restylefloat{figure}
```

in the preamble. Now you will be able to specify `[H]` as the position parameter, which will mean “HERE and only HERE”. This may cause an unwanted page break however¹⁵. If you want to avoid the unwanted pagebreak, i.e. let L^AT_EX move the float only if it doesn’t fit on the page, the use the `afterpage` package with:

```
\afterpage{\clearpage \begin{figure}[H] ... \end{figure}}
```

Complaints from L^AT_EX about “Too many floats” are usually caused by one of the above problems: floats not being able to be placed and L^AT_EX collecting too many of them. The solutions given above, especially those with `\clearpage` in them will usually help. In some cases there really are too many floats, as L^AT_EX has a limited number of “boxes” to store the floats. The package `morefloats` can be used to increase this number. If you need still more then you must edit a private copy of this file, but even then there will be some limit that you cannot pass. Then your only resort will be to change your document.

22 Multipage Floats

L^AT_EX’s floats cannot be split across pages. Sometimes, however, you want to have a table or figure that doesn’t fit on one page. The easiest way is to split these into multiple table or figure environments, but this has a number of undesirable effects:

- Where do you split it? This is in general more difficult for tables than for figures.
- How do you keep them together?
- You don’t want more than one entry in the list of figures/tables.

Although these problems are not fully solvable in all cases, here are a couple of suggestions:

22.1 Tables

For tables longer than a page you can use the `longtable` package. This package defines a `longtable` environment that is a kind of amalgamation of `table` and `tabular`. It has approximately the same syntax as the `tabular` environment, but it adds some features of `table`, like captions. Longtables will be automatically split when they don’t fit on the page. And they will be entered in the list of tables when a caption is given. They will not float, however, and cannot be used inside a float environment. This could mean that another `table` environment, which was defined before the `longtable`, will float past it, and therefore the numbers may get out of order. Another problem could be that the `longtable` starts rather far down the page, which isn’t a pleasant sight. If you want the `longtable` to start at the top of the page, the best thing to do is to include it in an `\afterpage` command (using the `afterpage` package). As a `longtable` is by definition large, it is best to put it in a separate file, and `\input` it in the `\afterpage` command:

```
\afterpage{\input{mytable}}
\afterpage{\clearpage\input{mytable}}
```

The last form has the additional advantage that most of the outstanding floats will be printed first.

22.2 Figures

There isn’t an equivalent `longfigure` solution, so for figures you will usually have to split it yourself. In general this is less of a problem. However, the problem you get now is how to keep them together, i.e. how to get the parts on subsequent pages, and how to get a single entry in the list of figures.

You will have to split the figure into pieces and put each part in a separate `figure` environment. To keep them together it is best to use only the `[p]` placement, so that they will be put on floatpages. As they are bigger than a page this is appropriate. The first part would then get a `\caption`, the subsequent parts would be used without a caption. If you want to add a caption-like text, enter it as normal text rather than a `\caption`, so that it will not be entered in the list of figures. It may also be desirable to issue a `\clearpage` first, just like we did for the `longtable`, and to encapsulate this in the `\afterpage` command. E.g.

¹⁴This command and the “!” placement parameter are not defined in L^AT_EX2.09.

¹⁵There used to be a `here.sty` with the same effect, but this is incompatible with L^AT_EX2_ε

```
\afterpage{\clearpage\input{myfigure}}
```

where `myfigure.tex` contains:

```
\begin{figure}[p]
\includegraphics{myfig1.eps}
\caption{This is a multipage figure}
\label{fig:xxx}
\end{figure}
\begin{figure}[p]
\includegraphics{myfig2.eps}
\begin{center}
Figure~\ref{fig:xxx} (continued)
\end{center}
\end{figure}
```

You have to make sure that the last part is big enough, otherwise L^AT_EX could decide to postpone it until it has collected some more floats. This can be done either by making the figure big enough (e.g. by adding some `\vspace`), or by tweaking the `\floatpagefraction` parameter.

If you want your multipage figure to start at a lefthand-side (even-numbered) page you can use a test in the `\afterpage` command (using the `ifthen` package):

```
\afterpage{\clearpage
\ifthenelse{\isodd{\value{page}}}{\afterpage{\input{myfigure}}} % odd page
{\input{myfigure}}} % even page
```

If there are too many floats on the skipped page, this may still fail to start your multipage figure on an even page, however.

23 Contact information

Piet van Oostrum
 Dept. of Computer Science
 Utrecht University
 P.O. Box 80.089
 3508 TB Utrecht, The Netherlands
 Telephone: +31 30 2531806 Telefax: +31 30 2513791
 E-mail: piet@cs.ruu.nl
 WWW: <http://www.cs.ruu.nl/people/piet>

Colored Verbatim

A vivid look at T_EX

Hans Hagen

april 18 1996

Abstract

This module implements (just another) verbatim environment. Especially when the output of T_EX is viewed on an electronic medium, coloring has a positive influence on the readability of T_EX sources. About half of the module is therefore dedicated to typesetting T_EX specific character sequences in color. In this article I'll present some macro's for typesetting inline, display and file verbatim. The macro's are capable of handling *<tabs>* too.

At PRAGMA we use the integrated environment T_EXEDIT for editing and processing T_EX documents. This program also supports real time spell checking and T_EX based file management. Although definitely not exclusive, the programs cooperate nicely with CONTEX_T, an integrated, parameter driven macro package that covers most of the things we want T_EX to do. Although T_EX can be considered a tool for experts, we've tried to put as less a burden on non-technical users as possible. This is accomplished in the following ways:

- We've added some trivial symmetry checking to T_EXEDIT. Sources are checked for the use of brackets, braces, begin-end and start-stop like constructions, with or without arguments.
- Although T_EX is very tolerant to unformatted input, we stimulate users to make the ASCII source as clean as possible. Many sources I've seen in distribution sets look so awful, that I sometimes wonder how people get them working. In our opinion, a good-looking source leads to less errors.
- We use parameter driven setups and make the commands as tolerant as possible. We don't accept commands that don't look nice in ASCII.
- Finally —I could have added some more— we use color.

When in spell-checking-mode, the words spelled correctly are shown in *green*, the unknown or wrongly spelled words are in *red* and upto four categories of words, for instance passive verbs and nouns, become *blue* (cyan) or *yellow*. Short and nearly always correct words are in white (on a black screen). This makes checking-on-the-fly very easy and convenient, especially because we place the accents automatically.

In T_EX-mode we show T_EX-specific stuff in appropriate colors and again we use four colors. We use those colors in a way that supports parameter driven setups, table typesetting and easy visual checking of symmetry. Furthermore the text becomes more readable.

color	characters that are influenced
red	{ } \$
green	\this \!!that \??these \@@those
yellow	` ' ~ ^ _ & / + - %
blue	() # [] " < > =

Macro-definition and style files often look quite green, because they contain many calls to macros. Pure text files on the other hand are mostly white (on the screen) and color clearly shows their structure.

When I prepared the interactive PDF manuals of CONTEX_T, T_EXEDIT and PPCHT_EX, I decided to include the original source text of the manuals as an appendix. At every chapter or (sub)section the reader can go to the corresponding line in the source, just to see how things were done in T_EX. Of course, the reader can jump from the source to corresponding typeset text too.

Confronted with those long (boring) sources, I decided that a colored output, in accordance with T_EXEDIT would be nice. It would not only visually add some quality to the manual, but also make the sources more readable.

Apart from a lot of *<catcode>*-magic, the task at hand was surprisingly easy. Although the macro's are hooked into the standard CONTEX_T verbatim mechanism, they are set up in a way that embedding them in another verbatim environment is possible.

This module includes part of the `CONTEXT` verbatim environment too, because it shows a few tricks that are often overseen by novice, especially the use of the `TeX` primitive `\meaning`. First I'll show in what way the users are confronted with verbatim typesetting. Because we want to be able to test for symmetry and because we hate the method of closing down the verbatim mode with some strange active character, we use the following construction for display verbatim:

```
\starttyping
the Dutch word 'typen' stands for 'typing', therefore the Dutch
implementation is in fact \starttypen ... \stoptypen
\stoptyping
```

Files can be typed with `\typefile` and inline verbatim can be accomplished with `\type`. This last command comes in two flavors:

We can say `\type<<something>>` or `\type{something}`. The first one is a bit longer but also supports slanted typing, which accomplished by typing `\type<<a <<slanted>> word>>`. We can also use commands to enhance the text `\type<<with <</bf boldfaced>> text>>`. Just to be complete, we decided to accept also `\LaTeX` alike verbatim, which means that `\type+something+` and `\type|something|` are valid commands too.

These commands can be tuned with accompanying setup commands. We can enable commands, slanted typing, control spaces, *(tab)*-handling and (here we are:) coloring. We can also setup surrounding white space and indenting.

I only present the framework macro's here, because the `CONTEXT`-setup command uses specific interface macros.¹ Embedding is up to the user.

The definition part of this text is typeset in color or greyscale. One has to keep in mind that the purpose of these macros was viewing `TeX` on an electronic medium. On paper, the results can be disappointing, because the quality depends on the printer. We start with some general macro's, some of which are only defined if they are `\undefined`.

```
1 1 \chardef\escapecode=0      \chardef\begingroupcode=1
2 2 \chardef\lettercode=11   \chardef\endgroupcode=2
3 3 \chardef\activecode=13
4 4 \def\zeropoint{0pt}
5 5 \ifx\scratchcounter\undefined \newcount\scratchcounter \fi
6 6 \ifx\everyline\undefined     \newtoks\everyline      \fi
7 7 \ifx\tempreadfile\undefined  \newread\tempreadfile \fi
8 8 \ifx\verbatimfont\undefined  \def\verbatimfont{\tt} \fi
4 9 \newif\ifitsdone
```

The inline verbatim commands presented here are a subset of the `CONTEXT` ones. Both grouped and character bound alternatives are provided. This command takes one argument: the closing command:

```
\processinlineverbatim{\closingcommand}
```

One can define his own verbatim commands, which can be very simple:

```
\def\Verbatim{\processinlineverbatim\relax}
```

or more complex:

```
\def\GroupedVerbatim%
{\bgroup
 \dosomeusefullthings
 \processinlineverbatim\egroup}
```

Before entering inline verbatim mode, we take care of the unwanted *(tabs)*, *(newlines)* and *(newpages)* (form feeds) and turn them into *(space)*. We need the double `\bgroup` construction to keep the closing command local.

```
5 10 \def\setupinlineverbatim%
11  {\verbatimfont
12  \let\obeytabs=\ignoretabs
13  \let\obeylines=\ignorelines
14  \let\obeypages=\ignorepages
15  \setupcopyverbatim}
```

¹At the moment `CONTEXT` has a Dutch interface. One of our targets is to fully document the source and make it public. As can be seen in the `PPCHTeX`-distribution, the underlying macros permit a multilingual interface, so we'll probably come up with an English version someday.

```

6 16 \def\doprocessinlineverbatim%
17   {\ifx\next\bgroup
18     \setupinlineverbatim
19     \catcode'\{=\begingroupcode
20     \catcode'\}=\endgroupcode
21     \def\next{\let\next=}
22   \else
23     \setupinlineverbatim
24     \def\next##1{\catcode'##1=\endgroupcode}%
25   \fi
26   \next}

7 27 \def\processinlineverbatim#1%
28   {\bgroup
29   \def\endofverbatimcommand{#1\egroup}%
30   \bgroup
31   \aftergroup\endofverbatimcommand
32   \futurelet\next\doprocessinlineverbatim}

```

The closing command is executed afterwards as an internal command and therefore should not be given explicitly when typesetting inline verbatim.

We can define a display verbatim environment with the command `\processdisplayverbatim` in the following way:

```
\processdisplayverbatim{\closingcommand}
```

For instance, we can define a simple command like:

```
\def\BeginVerbatim {\processdisplayverbatim{EndVerbatim}}
```

But we can also do more advance things like:

```
\def\BeginVerbatim {\bigskip \processdisplayverbatim{\EndVerbatim}}
\def\EndVerbatim {\bigskip}
```

When we compare these examples, we see that the backslash in the closingcommand is optional. One is free in actually defining a closing command. If one is defined, the command is executed after ending verbatim mode.

```

8 33 \def\processdisplayverbatim#1%
34   {\par
35   \bgroup
36   \escapechar=-1
37   \xdef\verbatimname{\string#1}%
38   \egroup
39   \def\endofdisplayverbatim{\csname\verbatimname\endcsname}%
40   \bgroup
41   \parindent\zeropoint
42   \ifdim\lastskip<\parskip
43     \remove\lastskip
44     \vskip\parskip
45   \fi
46   \parskip\zeropoint
47   \processingverbatimtrue
48   \expandafter\let\csname\verbatimname\endcsname=\relax
49   \edef\endofverbatimcommand{\csname\verbatimname\endcsname}%
50   \edef\endofverbatimcommand{\meaning\endofverbatimcommand}%
51   \verbatimfont
52   \setupcopyverbatim
53   \let\doverbatimline=\relax
54   \copyverbatimline}

```

The closing is saved in `\endofverbatimcommand` in such a way that it can be compared on a line by line basis. For the conversion we use `\meaning`, which converts the line to non-expandable tokens. We reset `\parskip`, because we don't want inter-paragraph skips to creep into the verbatim source. Furthermore we `\relax` the line-processing macro while getting the rest of the first line. The initialization command `\setupcopyverbatim` does just what we expect it to do: give all characters (*catcode*) 11. Furthermore we switch to french spacing and call for obedience.

```

9 55 \def\setupcopyverbatim%
56   {\uncatcodecharacters
57   \frenchspacing}

```

```

58 \obeyspaces
59 \obeytabs
60 \obeylines
61 \obeycharacters}

```

As its name says, `\uncatcodecharacters` resets the *⟨catcode⟩* of characters. Because we use an upper bound of 127, characters with higher values are not taken into account. When one wants to do special things with higher characters, this macro should be adapted.

```

10 62 \def\uncatcodecharacters%
63   {\scratchcounter=0
64    \loop
65     \catcode\scratchcounter=\lettercode
66     \advance\scratchcounter by 1
67     \ifnum\scratchcounter<127
68     \repeat}

```

We follow Knuth in naming macros that make *⟨space⟩*, *⟨newline⟩* and *⟨newpage⟩* active and assigning them `\obeysomething`. Their assigned values are saved in `\obeyedvalue`.

```

11 69 \def\obeyedspace {\hbox{ }}
70 \def\obeyedtab {\obeyedspace}
71 \def\obeyedpage {\vfill\eject}
72 \def\obeyedline {\par}

```

First we define `\obeyspaces`. When we want visible spaces (control spaces) we only have to adapt the definition of `\obeyedspace`:

```

12 73 \def\controlspace {\hbox{\char32}}
13 74 \bgroup
75 \catcode\ =\activecode
76 \gdef\obeyspaces{\catcode\ =\activecode\def {\obeyedspace}}
77 \gdef\setcontrolspaces{\catcode\ =\activecode\def {\controlspace}}
78 \egroup

```

Next we take care of *⟨newline⟩* and *⟨newpage⟩* and because we want to be able to typeset listings that contain *⟨tabs⟩*, we have to handle those too. Because we have to redefine the *⟨newpage⟩* character locally, we redefine the meaning of this (often already) active character.

```

14 79 \catcode\^^L=\activecode \def^^L{\par}
15 80 \bgroup
81 \catcode\^^I=\activecode
82 \catcode\^^M=\activecode
83 \catcode\^^L=\activecode
16 84 \gdef\obeytabs {\catcode\^^I=\activecode\def^^I{\obeyedtab}}
85 \gdef\obeylines {\catcode\^^M=\activecode\def^^M{\obeyedline}}
86 \gdef\obeypages {\catcode\^^L=\activecode\def^^L{\obeyedpage}}
17 87 \gdef\ignoretabs {\catcode\^^I=\activecode\def^^I{\obeyedspace}}
88 \gdef\ignorelines {\catcode\^^M=\activecode\def^^M{\obeyedspace}}
89 \gdef\ignorepages {\catcode\^^L=\activecode\def^^L{\obeyedline}}
18 90 \gdef\obeycharacters{}
19 91 \gdef\settabskips%
92   {\let\processverbatimline=\doprocstabskipline%
93    \catcode\^^I=\activecode\let^^I=\doprocstabskip}
20 94 \egroup

```

The main copying routine of `display verbatim` does an ordinary string-compare on the saved closing command and the line at hand. The space after #1 in the definition of `\next` is essential! As a result of using `\obeylines`, we have to use `%`'s after each line but none after the first #1.

```

21 95 {\obeylines%
96 \gdef\copyverbatimline#1
97   {\ifx\doverbatimline\relax% gobble rest of the first line
98    \let\doverbatimline=\dodoverbatimline%
99    \def\next{\copyverbatimline}%
100   \else%

```

```

101     \def\next{#1 }%
102     \ifx\next\emptyspace%
103         \def\next%
104             {\doemptyverbatimline{#1}%
105              \copyverbatimline}%
106     \else%
107         \edef\next{\meaning\next}%
108     \ifx\next\endofverbatimcommand%
109         \def\next%
110             {\egroup\endofdisplayverbatim}%
111     \else%
112         \def\next%
113             {\doverbatimline{#1}%
114              \copyverbatimline}%
115     \fi%
116 \fi%
117 \fi%
118 \next}}

```

The actual typesetting of a line is done by a separate macro, which enables us to implement *<tab>* handling. The trick with `\do` and `\dodo` enables us to obey the preceding `\parskip`, while skipping the rest of the first line. The `\relax` is used as an signal.

```

22 119 \def\dodoverbatimline#1%
120     {\leavevmode\the\everyline\strut\processverbatimline{#1}%
121      \everypar{ }%
122      \obeyedline\par}

```

Empty lines in verbatim can lead to white space on top of a new page. Because this is not what we want, we turn them into vertical skips. This default behavior can be overruled by:

```
\obeyemptylines
```

Although it would cost us only a few lines of code, we decided not to take care of multiple empty lines. When a (display) verbatim text contains more successive empty lines, this probably suits some purpose.

```

23 123 \bgroup
124 \catcode'\^^L=\activecode \gdef\emptypage {^^L}
125 \catcode'\^^M=\activecode \gdef\emptyline {^^M}
126                                     \gdef\emptyspace { }
127 \egroup
24 128 \def\doemptyverbatimline%
129     {\vskip\ht\strutbox
130      \vskip\dp\strutbox
131      {\setbox0=\hbox{\the\everyline}}}}
25 132 \def\obeyemptylines%
133     {\def\doemptyverbatimline{\doverbatimline}}

```

Although every line is a separate paragraph, we execute `\everypar` only once. In `CONTEXT` we use a bit different approach, because there we use `\everypar` for sidefloats, columnfloats and other features. We offer an alternative `\EveryPar`, which stacks `everypar`'s, while leaving the old one intact. For the same reason we implemented `\EveryLine`, which enables us to do things like line numbering while retaining `\everyline` behavior. Some other useful but distracting options have been removed here too.

We still have to take care of the *<tabs>*. A *<tab>* takes eight spaces and a *<space>* normally has a width of 0.5em. because we can be halfway a tabulation, we must keep track of the position. This takes time, especially when we print complete files, therefore we `\relax` this mechanism by default.

```

26 134 \def\doprocessstabskip%
135     {\obeyedspace % \hskip.5em\relax
136      \ifitsdone
137          \advance\scratchcounter 1\relax
138          \let\next=\doprocessstabskip
139          \itsdonefalse
140      \else\ifnum\scratchcounter>7\relax
141          \let\next=\relax
142      \else

```

```

143     \advance\scratchcounter 1\relax
144     \let\next=\doprocessstabskip
145     \fi\fi
146     \next}

27 147 \def\dodoprocessstabskipline#1#2\endoftabskipping%
148     {\ifnum\scratchcounter>7\relax
149       \scratchcounter=1\relax
150       \itsdonetrue
151     \else
152       \advance\scratchcounter 1\relax
153       \itsdonefalse
154     \fi
155     \ifx#1\relax
156       \let\next=\relax
157     \else
158       \def\next{#1\dodoprocessstabskipline#2\endoftabskipping}%
159     \fi
160     \next}

28 161 \let\endoftabskipping=\relax
162 \let\processverbatimline=\relax

29 163 \def\doprocessstabskipline#1%
164     {\bgroup
165       \scratchcounter=1\relax
166       \dodoprocessstabskipline#1\relax\endoftabskipping
167     \egroup}

```

The verbatim typesetting of files is done on a bit different basis. This time we don't check for a closing command, but look for *(eof)* and make sure it does not turn into an empty line.

```
\processfileverbatim{filename}
```

Typesetting a file in most cases results in more than one page. Because we don't want problems with files that are read in during the construction of the page, we declare `\ifprocessingverbatim`, so the output routine can adapt its behavior.

```

30 168 \newif\ifprocessingverbatim

31 169 \def\processfileverbatim#1%
170     {\par
171       \bgroup
172       \parindent\zeropoint
173       \ifdim\lastskip<\parskip
174         \removelastskip
175         \vskip\parskip
176       \fi
177       \parskip\zeropoint
178       \processingverbatimtrue
179       \uncatcodecharacters
180       \verbatimfont
181       \frenchspacing
182       \obeyspaces
183       \obeytabs
184       \obeylines
185       \obeypages
186       \obeycharacters
187       \openin\tempreadfile=#1%
188       \def\doreadline%
189         {\read\tempreadfile to \next
190           \ifeof\tempreadfile
191             % we don't want <eof> to be treated as <crlf>
192           \else\ifx\next\emptyline
193             \expandafter\doemptyverbatimline\expandafter{\next}%
194           \else\ifx\next\emptypage
195             \expandafter\doemptyverbatimline\expandafter{\next}%
196           \else
197             \expandafter\dodoverbatimline\expandafter{\next}%

```

```

198     \fi\fi\fi
199     \readline}%
200   \def\readline%
201     {\ifeof\tempreadfile
202       \let\next=\relax
203     \else
204       \let\next=\doreadline
205     \fi
206     \next}%
207   \readline
208   \closein\tempreadfile
209   \egroup
210   \ignorespaces}

```

These macro's can be used to construct the commands we mentioned in the beginning of this article. We leave this to the fantasy of the reader and only show some Plain T_EX alternatives for display verbatim and listings. We define three commands for typesetting inline text, display text and files verbatim. The inline alternative also accepts L^AT_EX-like verbatim.

```

\type{text}

\starttyping
... verbatim text ...
\stoptyping

\typefile{filename}

```

We can turn on the options by:

```

\controlspacetrue
\verbatimabstrue
\prettyverbatimtrue

```

Here is the implementation:

```

32 211 \newif\ifcontrolspace
    212 \newif\ifverbatimabs
    213 \newif\ifprettyverbatim
33 214 \def\presetting%
    215   {\ifcontrolspace
    216     \let\obeyspace=\setcontrolspace
    217   \fi
    218   \ifverbatimabs
    219     \let\obeytabs=\settabskips
    220   \fi
    221   \ifprettyverbatim
    222     \let\obeycharacters=\setupprettytextype
    223   \fi}
34 224 \def\type%
    225   {\bgroup
    226     \presetting
    227     \processinlineverbatim{\egroup}}
35 228 \def\starttyping%
    229   {\bgroup
    230     \presetting
    231     \processdisplayverbatim{\stoptyping}}
36 232 \def\stoptyping%
    233   {\egroup}
37 234 \def\typefile#1%
    235   {\bgroup
    236     \presetting
    237     \processfileverbatim{#1}%
    238   \egroup}

```

One can use the different `\obeysomething` commands to influence the behavior of these macro's. We use for instance `\obeycharacters` for making / an active character when we want to include typesetting commands.

We'll spend the remainder of this article on coloring the verbatim text.² We can turn on coloring by reassigning `\obeycharacters`:

```
\let\obeycharacters=\setupprettytextype
```

During pretty typesetting we can be in two states: *command* and *parameter*. The first condition becomes true if we encounter a backslash, the second state is entered when we meet a #.

```
38 239 \newif\ifintexcommand
240 \newif\ifintexparameter
```

The mechanism described here, is meant to be used with `color`. It is nevertheless possible to use different fonts instead of distinctive colors. When using `color`, it's better to end parameter mode after the #. When on the other hand we use a slanted typeface for the hashmark, then a slanted number looks better.

```
39 241 \newif\ifsplittexparameters \splittexparameterstrue
242 \newif\ifsplittexcontrols \splittexcontrolstrue
```

With `\splittexcontrols` we can influence the way control characters are processed in macronames. By default, the `^^` part is uncolored. When this boolean is set to false, they get the same color as the other characters.

The next boolean is used for internal purposes only and keeps track of the length of the name. Because two-character sequences starting with a backslash are always seen as a command.

```
40 243 \newif\iffirstintexcommand
```

We use a maximum of four colors because more colors will distract too much. In the following table we show the logical names of the colors, their color and rgb-values.

identifier	color	r	g	b	bw
texcolorone	red	0.9	0.0	0.0	0.30
texcolortwo	green	0.0	0.8	0.0	0.45
texcolorthree	yellow	0.0	0.0	0.9	0.60
texcolorfour	blue	0.8	0.8	0.6	0.75

This following poor mans implementation of color is based on PostScript. One can of course use grayscales too.

```
41 244 \def\setcolorverbatim%
245   {\splittexparameterstrue
246   \def\texcolorone { .9 .0 .0 } % red
247   \def\texcolortwo { .0 .8 .0 } % green
248   \def\texcolorthree { .0 .0 .9 } % blue
249   \def\texcolorfour { .8 .8 .6 } % yellow
250   \def\texbeginofpretty[##1]%
251   {\special{ps:: \csname##1\endcsname setrgbcolor}}
252   \def\texendofpretty%
253   {\special{ps:: 0 0 0 setrgbcolor}} % black

42 254 \def\setgrayverbatim%
255   {\splittexparameterstrue
256   \def\texcolorone { .30 } % gray
257   \def\texcolortwo { .45 } % gray
258   \def\texcolorthree { .60 } % gray
259   \def\texcolorfour { .75 } % gray
260   \def\texbeginofpretty[##1]%
261   {\special{ps:: \csname##1\endcsname setgray}}
262   \def\texendofpretty%
263   {\special{ps:: 0 setgray}} % black
```

One can redefine these two commands after loading this module. If available, one can use appropriate font-switch macro's. We default to color.

```
43 264 \setcolorverbatim
```

Here come the commands that are responsible for entering and leaving the two states. As we can see, they've got much in common.

```
44 265 \def\texbeginofcommand%
266   {\texendofparameter
```

²The original macro's have some primitive symmetry testing options.

```

267 \ifintexcommand
268 \else
269 \global\intexcommandtrue
270 \global\firstintexcommandtrue
271 \texbeginofpretty[texcolortwo]%
272 \fi}
45 273 \def\texendofcommand%
274 {\ifintexcommand
275 \texendofpretty
276 \global\intexcommandfalse
277 \global\firstintexcommandfalse
278 \fi}
46 279 \def\texbeginofparameter%
280 {\texendofcommand
281 \ifintexparameter
282 \else
283 \global\intexparametertrue
284 \texbeginofpretty[texcolorthree]%
285 \fi}
47 286 \def\texendofparameter%
287 {\ifintexparameter
288 \texendofpretty
289 \global\intexparameterfalse
290 \fi}

We've got nine types of characters. The first type concerns the grouping characters that become red and type seven takes care of the backslash. Type eight is the most recently added one and handles the control characters starting with ^^ . In the definition part at the end of this article we can see how characters are organized by type.

48 291 \def\ifnotfirstintexcommand#1%
292 {\iffirstintexcommand
293 \string#1%
294 \texendofcommand
295 \else}
49 296 \def\textypeone#1%
297 {\ifnotfirstintexcommand#1%
298 \texendofcommand
299 \texendofparameter
300 \texbeginofpretty[texcolorone]\string#1\texendofpretty
301 \fi}
50 302 \def\textyepetwo#1%
303 {\ifnotfirstintexcommand#1%
304 \texendofcommand
305 \texendofparameter
306 \texbeginofpretty[texcolorthree]\string#1\texendofpretty
307 \fi}
51 308 \def\textyepethree#1%
309 {\ifnotfirstintexcommand#1%
310 \texendofcommand
311 \texendofparameter
312 \texbeginofpretty[texcolorfour]\string#1\texendofpretty
313 \fi}
52 314 \def\textyepfour#1%
315 {\ifnotfirstintexcommand#1%
316 \texendofcommand
317 \texendofparameter
318 \string#1%
319 \fi}
53 320 \def\textyepfive#1%
321 {\ifnotfirstintexcommand#1%
322 \texbeginofparameter
323 \string#1%

```



```

324     \fi}
54 325 \def\texttypesix#1%
326     {\ifnotfirstintexcommand#1%
327         \ifintexparameter
328             \ifsplittexparameters
329                 \texendofparameter
330                 \string#1%
331             \else
332                 \string#1%
333             \texendofparameter
334         \fi
335     \else
336         \texendofcommand
337         \string#1%
338     \fi
339     \fi}
55 340 \def\textypeseven#1%
341     {\ifnotfirstintexcommand#1%
342         \texbeginofcommand
343         \string#1%
344     \fi}
56 345 \def\dodotextypeeight#1%
346     {\texendofparameter
347     \ifx\next#1%
348         \ifsplittexcontrols
349             \ifintexcommand
350                 \texendofcommand
351                 \string#1\string#1%
352             \texbeginofcommand
353         \else
354             \string#1\string#1%
355         \fi
356     \else
357         \string#1\string#1%
358     \fi
359     \let\next=\relax
360 \else
361     \textypethree#1%
362 \fi
363 \next}
57 364 \def\textypeeight#1%
365     {\def\dotextypeeight{\dodotextypeeight#1}%
366     \afterassignment\dotextypeeight\let\next=}
58 367 \def\textypenine#1%
368     {\texendofparameter
369     \global\firstintexcommandfalse
370     \string#1}

```

We have to take care of the control characters we mentioned before. We obey their old values but only after ending our two states.

```

59 371 \def\texsetcontrols%
372     {\global\let\oldobeyedspace=\obeyedspace
373     \global\let\oldobeyedline=\obeyedline
374     \global\let\oldobeyedpage=\obeyedpage
375     \def\obeyedspace%
376         {\texendofcommand
377         \texendofparameter
378         \oldobeyedspace}%
379     \def\obeyedline%
380         {\texendofcommand
381         \texendofparameter
382         \oldobeyedline}%

```

```

383 \def\obeyedpage%
384 {\texendofcommand
385 \texendofparameter
386 \oldobeyedpage}}

```

Next comes the tough part. We have to change the *(catcode)* of each character. These macro's are tuned for speed and simplicity. When viewed in color they look quite simple.

```

60 387 \def\setupprettytextype%
388 {\texsetcontrols
389 \texsetspecialpretty
390 \texsetalphabetpretty
391 \texsetextrapretty}

```

When handling the lowercase characters, we cannot use lowercased macro names. This means that we have to redefine some well known macros, like `\bgroup`.

```

61 392 \def\textpresetcatcode%
393 {\def\##1%
394 {\expandafter\catcode\expandafter'\csname##1\endcsname\activecode}}

62 395 \def\textsettypenine%
396 {\def\##1%
397 {\def##1{\textypenine##1}}}

63 398 \bgroup
399 \bgroup
400 \gdef\textpresetalphapretty%
401 {\textpresetcatcode
402 \A\B\C\D\E\F\G\H\I\J\K\L\M%
403 \N\O\P\Q\R\S\T\U\V\W\X\Y\Z}
404 \textpresetalphapretty
405 \gdef\textsetalphapretty%
406 {\textpresetalphapretty
407 \textsettypenine
408 \A\B\C\D\E\F\G\H\I\J\K\L\M%
409 \N\O\P\Q\R\S\T\U\V\W\X\Y\Z}
410 \egroup
411 \global\let\TEXPRESETCATCODE = \textpresetcatcode
412 \global\let\TEXSETTYPENINE = \textsettypenine
413 \global\let\BGROUP = \bgroup
414 \global\let\EGROUP = \egroup
415 \global\let\GDEF = \gdef
416 \BGROUP
417 \GDEF\TEXPRESETALPHAPRETTY%
418 {\TEXPRESETCATCODE
419 \a\b\c\d\e\f\g\h\i\j\k\l\m%
420 \n\o\p\q\r\s\t\u\v\w\x\y\z}
421 \TEXPRESETALPHAPRETTY
422 \GDEF\TEXSETALPHAPRETTY%
423 {\TEXPRESETALPHAPRETTY
424 \TEXSETTYPENINE
425 \a\b\c\d\e\f\g\h\i\j\k\l\m%
426 \n\o\p\q\r\s\t\u\v\w\x\y\z}
427 \EGROUP
428 \gdef\textsetalphabetpretty%
429 {\textsetalphapretty
430 \TEXSETALPHAPRETTY}
431 \egroup

```

Macronames normally only contain characters. As mentioned before, we also permit the characters @, ! and ?. Of course they are only colored (green) when they are part of the name.

```

64 432 \bgroup
433 \gdef\textpresetextrapretty%
434 {\textpresetcatcode
435 \!\@\}
436 \textpresetextrapretty

```

```

437 \gdef\texsetextrapretty%
438   {\texpresetextrapretty
439    \texsettypenine
440    \!\!\!\!\@}
441 \egroup

```

Here comes the main linking routine. In this macro we also have to change the escape character to ! and use X, Y and Z for grouping and ignoring, which makes the result a bit less readable. Plain TeX defines \+ as an outer macro, so we have to redefine this one too.

```

65 442 \def\+{\tabalign} % Plain TeX: \outer\def\+{\tabalign}
66 443 %<TeX_Marker>
444 \bgroup
445 \gdef\texpresetspecialpretty%
446   {\def\##1{\catcode\##1\activecode}%
447    \!\!\!\!\! = \!\!\!\!\! < \!\!\!\!\! > \!\!\!\!\! # \!\!\!\!\! / \!\!\!\!\! \!\!\!\!\! "%
448    \!\!\!\!\! $ \!\!\!\!\! { \!\!\!\!\! } %
449    \!\!\!\!\! - \!\!\!\!\! + \!\!\!\!\! | \!\!\!\!\! % \!\!\!\!\! \!\!\!\!\! _ \!\!\!\!\! ^ \!\!\!\!\! & \!\!\!\!\! ~ \!\!\!\!\! / \!\!\!\!\! ' %
450    \!\!\!\!\! . \!\!\!\!\! , \!\!\!\!\! : \!\!\!\!\! ; %
451    \!\!\!\!\! * %
452    \!\!\!\!\! 1 \!\!\!\!\! 2 \!\!\!\!\! 3 \!\!\!\!\! 4 \!\!\!\!\! 5 \!\!\!\!\! 6 \!\!\!\!\! 7 \!\!\!\!\! 8 \!\!\!\!\! 9 %
453    \!\!\!\!\! }
454    \catcode\X=\the\catcode\{
455    \catcode\Y=\the\catcode\}
456    \catcode\Z=\the\catcode\%
457 \gdef\texsetsometypes%
458   {\def\!##1##2{\def##1{##2{##1}}}%
459 XZ
460 \catcode\!=\escapecode
461 !texpresetspecialpretty
462 !gdef!texsetspecialpretty
463 XZ
464 !texpresetspecialpretty
465 !texsetsometypes
466 !! $ !textypeone !! { !textypeone !! } !textypeone
467 !! [ !textypetwo !! ] !textypetwo !! ( !textypetwo !! ) !textypetwo
468 !! = !textypetwo !! < !textypetwo !! > !textypetwo !! " !textypetwo
469 !! - !textypethree !! + !textypethree !! / !textypethree
470 !! | !textypethree !! % !textypethree !! ' !textypethree !! ` !textypethree
471 !! _ !textypethree !! ^ !textypethree !! & !textypethree !! ~ !textypethree
472 !! . !textypefour !! , !textypefour !! : !textypefour !! ; !textypefour
473 !! * !textypefour
474 !! # !textypefive
475 !! 1 !textypesix !! 2 !textypesix !! 3 !textypesix
476 !! 4 !textypesix !! 5 !textypesix !! 6 !textypesix
477 !! 7 !textypesix !! 8 !textypesix !! 9 !textypesix
478 !! \ !textypeseven
479 !! ^ !textypeeight
480 YZ
481 YZ
482 \egroup

```

This text is included in the file where the macro's are defined. In this article, the verbatim part of this text was set with the following commands for the examples:

```

\def\starttypen% We simplify the \ConTeXt\ macro.
{\bgroup
 \everypar{} % We disable some \ConTeXt\ / \LaTeX/ mechanisms.
 \advance\leftskip by 1em
 \processdisplayverbatim{\stoptypen}}

\def\stoptypen%
{\egroup}

```

The implementation itself was typeset with:

```

\def\startdefinition%
  {\bgroup
   \everypar{} % We disable some \ConTeXt\ / \LaTeX/ mechanisms.
   \let\obeycharacters=\setupprettytextype
   \everypar{\showparagraphcounter}%
   \everyline{\showlinecounter}%
   \verbatimcorps
   \processdisplayverbatim{\stopdefinition}}

\def\stopdefinition%
  {\egroup}

```

Because we have both `\everypar` and `\everyline` available, we have implemented a dual numbering mechanism:

```

\newcount\paragraphcounter
\newcount\linecounter

\def\showparagraphcounter%
  {\llap
   {\bgroup
    \counterfont
    \hbox to 4em
     {\global\advance\paragraphcounter by 1
      \hss \the\paragraphcounter \hskip2em}}%
   \egroup
   \hskip1em}}

\def\showlinecounter%
  {\llap
   {\bgroup
    \counterfont
    \hbox to 2em
     {\global\advance\linecounter by 1
      \hss \the\linecounter}}%
   \egroup
   \hskip1em}}

```

Of course commands like this have to be embedded in a decent setup structure, where options can be set at will.

Let's summarize the most important commands.

```

\processinlineverbatim{\closingcommand}
\processdisplayverbatim{\closingcommand}
\processfileverbatim{filename}

```

We can satisfy our own specific needs with the following interfacing macro's:

```
\obeyspaces \obeytabs \obeylines \obeypages \obeycharacters
```

Some needs are fulfilled already with:

```
\setcontrolspace \settabskips \setupprettytextype
```

lines can be enhanced with ornaments using:

```
\everypar \everyline
```

and color support is implemented by:

```
\texbeginofpretty[#1] ... \texendofpretty
```

We can influence the verbatim environment with the following macro and booleans:

```
\obeyemptylines \splittexparameters... \splittexcontrols...
```

This macro can be redefined by the user. The parameter #1 can be one of the four 'fixed' identifiers *texcolorone*, *texcolortwo*, *texcolorthree* and *texcolorfour*. We have implemented a more or less general PostScript color support mechanism, using specials. One can toggle between color and grayscale with:

```
\setgrayverbatim \setcolorverbatim
```

We did not mention one drawback of the mechanism described here. The closing command must start at the first position of the line. The original implementation does not have this drawback, because we test if the end command is a substring of the line at hand. Although the two macros that we use for this only take a few lines of code, we think they are out of place in this article.

One can wonder why such a simple application takes 482 lines of $\text{T}_{\text{E}}\text{X}$ -code. But then, $\text{T}_{\text{E}}\text{X}$ was never meant to be simple.