Nederlandstalige TEX Gebruikersgroep

Aan	:	Leden Nederlandstalige TEX Gebruikersgroep,
	:	Secretariaten TEX Gebruikersgroepen,
	:	Europese TEX coördinator.
Betreft	:	Verslag 5^{e} bijeenkomst Nederlandstalige TEX Gebruikersgroep
Locatie	:	Katholieke Universiteit Nijmegen
Datum	:	11 mei 1990
Behandeling	:	bij de volgende bijeenkomst (20 november 1990 te Utrecht)

Agenda:

Bijlagen:

De NTG vereniging

Voorzitter:	C.G. van der Laan, Rijksuniversiteit Groningen. Internet: cgl@rug.nl
Secretaris:	G.J.H. van Nes, ENR/ECN Rekencentrum, Petten. Internet: vannes@ecn.nl
Penningmeester:	J.L. Braams, PTT Dr Neher Laboratorium, Leidschendam. Internet: jl_braams@pttrnl.nl
Bestuursleden:	H.P.A. Mulders, KUB, Tilburg. T. de Klerk, DEC, Utrecht.
Postadres:	Nederlandstalige TEX Gebruikersgroep, Postbus 394, 1740 AJ Schagen.
Postgiro:	1306238, t.n.v. Penningmeester NTG, Zoetermeer.
Internet:	ntg@hearn

De Nederlandstalige TEX Gebruikersgroep (NTG) is een vereniging die tot doel heeft het bevorderen van de kennis en het gebruik van TEX.
De **NTG** tracht dat te bereiken door het uitwisselen van informatie, het organiseren van congressen, symposia en tentoonstellingen m.b.t. TEX en 'TEX-produkten', en door het onderzoeken en vergelijken van TEX met soortgelijke/aanverwante produkten, b.v. SGML.

De **NTG** biedt haar leden ondermeer het volgende:
- Tweemaal per jaar een **NTG**-bijeenkomst.
- Eenmaal per jaar open '**NTG**-dagen', waar naast lezingen, ook cursussen (speciaal tarief voor leden) worden gegeven.
- De fileserver TEX-NL waarop algemeen te gebruiken 'TEX-produkten' staan. De meeste van deze TEX-produkten zijn, tegen geringe vergoeding, ook op floppy verkrijgbaar. Daaronder valt ook een MS-DOS versie van TEX, LATEX, en een previewer.
- De discussielijst TEX-NL waarop vragen gesteld worden. Ook worden er via deze listserver ervaringen uitgewisseld.
- Aktiviteiten in werkgroepen.
- Eenmaal per jaar een ledenlijst met per lid informatie welke software en welke hardware, in relatie met TEX, wordt gebruikt.

Lid worden kan door overmaking aan de penningmeester van het verschuldigde contributie bedrag. Daarnaast dient een informatieformulier te worden ingevuld, welke laatste via het secretariaat te verkrijgen is.

De contributie voor een persoonlijk lidmaatschap bedraagt f 75,–, de contributie voor een instituutslidmaatschap bedraagt f 200,–. 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 50,– per persoon. Voor studenten geldt eveneens een tarief van f 50,- (geen stemrecht). Voor afwijkende regelingen dient contact met het bestuur opgenomen te worden.

De statuten van de Nederlandse TEX-Gebruikersgroep zijn via het secretariaat te verkrijgen.

Nederlandstalige TEX Gebruikersgroep

Aanwezig : A. Al-Dhahir (UT); E. Algera (EGD); A.W.W.M. Biegstraaten (TUD); J.L. Braams (PTT Neher Laboratorium); M.A.J.H. Broeren (Océ Nederland B.V.); H. Brouwer (EGD); P.C.M. van Campen (KUN); N. Cox (KUN); M. Dings (AKZO); W. van Dongen (Inter Documentation Company B.V.); R. Doornebal (Wolters Kluwer Academic Publishers); G.D. Draaijer (MARIN); V. Eijkhout (KUN); E. van Eynde (UR, België); E.J. Evers (RUU); J. van Gent (KUB); A.J. van der Goot (EGD); G. Haayer (Styx Publ.); B.H. Hasselman; P.E.M. Huygen (AZR); A.J. de Jong; W.J. Karman (KUN); T. de Klerk (DEC); J. van Krieken (KUN); G.D.C. Kuiken (KUIKEN); C.G. van der Laan (RUG); A. de Leeuw van Weenen (RUL); A. Lenstra; J. Maasdam (Intermedia bv/Samsom-Sijthoff); J.C. de Moor (Theologische Universiteit); H.P.A. Mulders (KUB); G.J.H. van Nes (ECN); G. Oomen (Wolters Kluwer); P. van Oostrum (RUU); B. Polman (KUN); N.A.F.M. Poppelier (Elsevier Science Publishers); C.H. de Ridder (Philips Components); A. Soos (Universiteit Twente); E.J. Vens (RUG); E.R. de Vreede (DCMR); J.J. Winnink; D. van Wijnen (Wolters Kluwer Academic Publishers).

Notulist : G.J.H. van Nes (ECN)

1 Opening

Voorzitter van der Laan (RUG) opent om 10:45 uur de bijeenkomst en memoreert dat sinds de eerste NTG bijeenkomst in juni 1988 er zeer veel is gebeurd. Na wat pionierswerk van enkele geïnteresseerde LaTeX gebruikers zijn er, naast het instellen van een voorlopig bestuur, diverse activiteiten ontwikkeld. NTG is een echte vereniging geworden met ruim 90 leden waaronder 15 instituutsleden. Daarnaast blijft de belangstelling voor de listserver `TeX-NL` stijgen.
De NTG heeft sinds kort ook een eigen postbus en het bestuur is te bereiken via het e-mail adres `NTG@HEARN`. Daarnaast is er vooral de laatste tijd veel contact met Europese zusterverenigingen.

De komende tijd zal aandacht geschonken moeten worden aan de *continuiteit van de vereniging*. Het huidige bestuur heeft een bestuursperiode van drie jaar op zich genomen. Vanaf juni 1991 zullen daarom ook jaarlijks verkiezingen gehouden worden, waarbij twee bestuursleden hun positie verkiesbaar stellen.
Behalve aan vernieuwing binnen het bestuur dient er ook aandacht aan de continuïteit van de werkgroepen besteed te worden: nieuwe mensen moeten ingewijd worden in de vergaarde kennis en de activiteiten van de groep.

Ook de *promoting van NTG en TEX* via de boekhandels (bijsluiten van informatie en PD software bij boeken en producten) wordt voorgesteld ten einde geïnteresseerden op een snelle manier voor te lichten. Gevraagd wordt om vrijwilligers.

Medegedeeld wordt dat de NTG vandaag te gast is bij 'Informatica' en 'Computer & Communicatie'.

De agenda zoals deze is gepresenteerd, wordt goedgekeurd.

2 Verslag bijeenkomst 23 november 1989

Betreffende het verslag van de 4^e NTG bijeenkomst te Tilburg, worden de volgende opmerkingen gemaakt:

- Bij de adressering 'Aan' (blz 1), dienen toegevoegd te worden de zusterverenigingen plus de Europese coördinator,
- Gewezen wordt op de mogelijkheid van het NTG studententarief (blz. 2),
- 'Japanese krant' (blz. 3) moet gewijzigd worden in 'Japanse krant',
- Poppelier deelt mede dat zijn adres bij RUU veranderd moet worden in TEXnique,
- De naam Tutelaers is niet altijd goed gespeld in het verslag,
- Het is beter om de naam 'stijloptie' i.p.v. 'stijlfile' te gebruiken (2^e regel blz. 8),
- BRIEF.TEX moet zijn: BRIEFDOC.TEX (4^e regel blz 8),

Het verslag wordt verder goedgekeurd (en zal worden getekend door voorzitter en secretaris).

This report has been generated by LaTeX software on SUN platform, using times.sty and multicol.sty (Frank Mittelbach), and outputdevice LN03R. Editor of this report is G.J.H. van Nes.

Betreffende de bijlagen wordt het volgende opgemerkt/medegedeeld:

- **Bijlage C:** De plaats van Bleeker wordt i.v.m. beëindiging dienstbetrekking overgenomen door Poppelier. Brouwer trekt zich als coördinator terug (niet als lid) uit werkgroep 5. De plaats is vacant.
- **Bijlage D:** Zowel de inkomsten als de uitgaven lijst bevat een fout in de optelling. Het bedrag *f* 16.168,10 dient te zijn: *f* 13.168,10.
- **Bijlage E:** Het uitgeven van een 'NTG rapporten serie' is binnen het bestuur besproken. Ondanks de vele voordelen, zal de uitvoering toch enige mankracht kosten. Voorlopig heeft het een lage prioriteit.
- **Bijlage F:** Medegedeeld wordt dat het aantal abonnees reeds de honderd gepasseerd is.
- **Bijlage K:** Maasdam (Intermedia) stelt voor deze werkgroep het eerste te behandelen bij het agendapunt 'werkgroepen'.
- **Bijlage L:** Betreffende deze discussie op TEX-NL wordt medegedeeld dat dit soort discussies vermeden moet worden op het netwerk. Men moet elkaar kunnen helpen zonder dat er discussie plaatsvindt in hoeverre een vraag relevant is. De discussie op het net bleek ook uiteindelijk niet zo bedoeld te zijn.
  Bij vragen dient, indien mogelijk, ook eerst een locale guru te worden geraadpleegd, voordat het netwerk ervoor gebruikt wordt. Er moet in het algemeen worden opgepast voor 'netwerkvervuiling'. TEX-NL kent geen moderator.
  De voorzitter benadrukt dat het zeer positief is, dat diverse NTG leden steeds de tijd hebben de voorkomende vragen via TEX-NL te beantwoorden. Hopelijk zal deze vorm van consultancy in de toekomst ook op basis van vrijwillegers kunnen blijven plaatsvinden.
- **Bijlage S:** Kort wordt besproken het nut van het opnemen van TUGboat artikelen (zoals ook genoemde bijlage) bij het NTG verslag. Daar slechts een beperkte groep van de NTG leden ook lid is van TUG, en dit soort bijdragen dikwijls pas later verschijnen in een TUGboat uitgave, wordt geconcludeerd dat dit soort (Nederlandse) bijdragen welkom blijven voor het NTG verslag.
- **Bijlage T:** Poppelier deelt mede dat hij vanaf begin 1990 bij Elsevier Science Publishers werkzaam is. Voor wat betreft de cursussen die hij geeft, geldt als adres echter het bedrijf TEXnique te Utrecht.
- **Bijlage U:** De naam Reiner S (paragraaf 'Toekomst LATEX') moet worden Rainer Schöpf. Voorgesteld wordt om ook geen afkortingen van namen meer te gebruiken.
- **Bijlage V:** Status over deze activiteiten is nog onbekend. reactie van Bart Childs is nog niet ontvangen. Mogelijk dat de enquête van Anita Hoover welke onlangs op het net was geplaatst, hiermee te maken had. De voorzitter zal op de GUTenberg bijeenkomst te Toulouse trachten hierover informatie te verkrijgen. Poppelier verzoek alle initialen van hem op te nemen (referenties).
- **Bijlage W:** Artikel is verschenen in het blad Tyroskoop waarvan een proefexemplaar op de vergadering aanwezig was. Een Engelstalige vertaling ervan is beschikbaar.

Genoemd worden enige suggesties voor bijlagen voor het volgende verslag. Zeker zal het verslag van de Cork'90 bijeenkomst worden opgenomen. Gedacht moet ook worden aan ondermeer: 'beginners informatie' (wat is beschikbaar, waar te verkrijgen), lijst van commercieel verkrijgbare software en het verzamelen van vragen op het gebied van communicatie.

De voorzitter deelt mede dat binnen het bestuur de vorm van het verslag ter discussie is gebracht. De huidige vorm: het verslag van de vergadering met een uitgebreide verzameling van bijlagen, zou omgezet kunnen worden in een verslag met separaat een tijdschrift. Aan de aanwezigen leden wordt de mening gevraagd. Genoemd wordt dat de huidige vorm een groot aantal voordelen heeft, doch dat het bestuur open staat voor suggesties van de leden.

3 Ingekomen stukken en Mededelingen

De volgende stukken zijn ingekomen:

- Tape (Cyber?) uit Los Alamos met diverse stijlfiles. Gevraagd wordt om een Cybersite (ECN?) waar deze tape gelezen kan worden.
- Verslagen van zusterverenigingen.
- Informatie over TUGLIB@math.utah.edu.

De volgende mededelingen worden gedaan:

- **Ledenlijst**
  De aanwezige worden gevraagd om correcties in de ledenlijst door te geven aan de secretaris.
- **SGML**
  Medegedeeld wordt dat SGML Users Group Holland een officiële vereniging is.

4 Verslag/discussie werkgroepen

Verslag wordt gedaan van de activiteiten binnen enkele NTG werkgroepen. De lijst met deelnemers van de werkgroepen is te vinden in *bijlage C*.
De volgorde van de werkgroepen in dit verslag is niet overeenkomend met de volgorde van behandeling gedurende de vergaderingen, welke laatste meer gebaseerd was op de graad van belangrijkheid.

Werkgroep 1: Cursusmateriaal

Betreffend cursusmateriaal wordt opgemerkt dat het zinvol zou zijn het nu aanwezige materiaal te bundelen en in rapport vorm uit te geven

Eijkhout (KUN) noemt een aantal nieuwe leerboeken op het gebied van TEX waarvan hij een recensie heeft ge-

maakt *(zie ook bijlage X)*. De Klerk (DEC) is zelf met een boek over TEX bezig. Van Eijkhout zal in 1991 ook een TEX boek verschijnen.

De voorzitter juicht locale activiteiten in de vorm van eigen cursussen toe. Benadrukt wordt de belangrijkheid van een centrale registratie van het materiaal wat allemaal aanwezig is. Mogelijk kan hier in de toekomst een bijlage aan worden besteed. Bijlage M van het vorige verslag (blz. 49 e.v.) besteedde hieraan reeds enige aandacht.

Werkgroep 2: Richtlijnen voor publicaties/conferenties proceedings etc.

De werkgroep is niet zo actief geweest. Wel is er voor Kluwer een rapport gemaakt. TUGboat stijlen zijn beschikbaar, goed gedocumenteerd en zeer leerzaam. In het algemeen wordt gesteld dat bij het maken van stijlen men van een DOC file moet uitgaan.

Gesproken wordt tevens over een voorzet voor een NTG rapporten stijl. Van Gent (KUB) zal samen met werkgroep 13 zich hierover buigen.

Werkgroep 3: Evaluatie produkten

Braams (PTT Neher Lab.) deelt mede dat de werkgroep zeer sluimerend is. Er is een WP naar LATEX converter op de TEX-NL fileserver geplaatst voor algemeen gebruik. Deze converter blijkt op een aantal punten niet geheel te voldoen. Voor een zeer groot gedeelte wordt de conversie goed uitgevoerd, doch handwerk blijft toch nog altijd vereist.
Op de vraag of er een rapport over de kwaliteit van de conversie software beschikbaar is, antwoordt Braams dat er alleen een Engelstalige handleiding beschikbaar is. Helmig (Philips) schijnt meer informatie te bezitten. De voorzitter vermeldt dat er bij de RUG enige tijd geleden K-Talk nader aan de tand is gevoeld. Naar aanleiding hiervan is er bij de makers van DRILCON (KUB) de testcollectie van de RUG gebruikt. Conclusie m.b.t. K-Talk was dat het efficienter is om direct in 'normale' ASCII tekst de invoer voor TEX te maken.
Kuiken (KUIKEN) meldt dat er nog een PD WordPerfect (5.0) conversie programma beschikbaar is. Pasmanter (Rijkswaterstaat/KNMI) zou er kennis van hebben.

Al-Dhahir (UT) zegt dat er ook belangstelling is voor een Chiwriter naar LATEX converter. Braams deelt mede dat bij zijn bedrijf iemand naar zo'n converter heeft gekeken. Echter de tendens is dat steeds meer TEX geleerd wordt i.p.v. Chiwriter, temeer daar uitwisseling van documenten steeds belangrijker wordt.

Poppelier (Elsevier) merkt op dat er een sterke behoefte is naar een grafische interface voor TEX.

Werkgroep 4: Fonts

Er was niets te melden.

Werkgroep 5: Drivers, previewers, printers, postscript

Er was niets te melden.

Werkgroep 6: Lijst en link met fotozetters

Er was niets te melden.

Werkgroep 7: PC-zaken

Vens (RUG) en Winnink willen zich bij de werkgroep aansluiten.
Winnink meldt dat hij enige activiteiten heeft ontwikkeld m.b.t. het onderzoek van de public domain DOS PC implementaties:

- PubTEX (slechte documentatie),
- DOSTEX (is traag) en
- SbTEX (ongeveer vijf maal sneller dan DOSTEX).

SbTEX is wel een kale distributieversie waarbij het LATEX deel niet zou draaien. Met minimaal 480 kBytes RAM geheugen is wel goed en snel met TEX te werken. Ook een PiCTEX implementatie werkt goed op de PC. Van de TRICKLE server is tevens een DVI naar VGA driver opgehaald, welke zeer goed blijkt te werken.
Winnink probeert uit genoemde drie TEX implementaties, een redelijk goed product te destilleren met minimaal TEX, LATEX, en mogelijk een previewer. DOSTEX valt daarbij zeker af daar bij deze implementatie tevens geheugenproblemen optreden.

Betreffende de DVI naar VGA driver heeft van Gent (KUB) dezelfde ervaringen. Bij het wisselen van de pagina's moet er een herformatering plaatsvinden hetgeen de nodige tijd kost.

Bison (KUN) meldt dat hij met de emTEX implementatie bezig is. Karman (KUN) meldt dat deze implementatie op INITEX na, volledig is
Van Oostrum (RUU) heeft net SbTEX gekregen.

Discussie wordt gevoerd over de vraag wat bij distributie onder een 'volledig systeem' wordt verstaan (inclusief makeindex?). Poppelier (Elsevier) meldt dat Van der Zalm (RUU) hiervan een lijst heeft gemaakt.

Betreffende de distributie van een volledige MS-DOS TEX versie zal Winnink nog contact opnemen met het bestuur.
Genoemd wordt de mogelijkheid van het maken van een 'batch-file' met de volledige installatie procedure. Benadrukt werd om alle suggesties betreffende de MS-DOS implementatie door te geven aan de coördinator van de PC-DOS werkgroep (Winnink).

Eijkhout (KUN) meldt dat Sonnemans (Atari ST nieuws) de 35 Atari diskettes van Matthias Moritz heeft overgenomen. Hopelijk kan daarbij een basis worden gelegd voor de TEX distributie voor de Atari systemen. Van

Oostrum (RUU) deelt mede dat hij een C versie van een server heeft gehaald inclusief volledige source.

Van der Laan (RUG) stelt voor om de distributie pas te beginnen als versie 3.0 beschikbaar is. Braams (PTT) meldt dat van SbTEX reeds versie 3.0 beschikbaar is.
Een oproep wordt gedaan voor β-testers. Winnink meldt dat hij in ieder geval α tester wil zijn.

Betreffende de Amiga systemen meldt Algera (EGD) dat hij net zo'n systeem in zijn bezit heeft en nu pas tot actie kan overgaan. Hij zal met Tutelaers (TUE) contact opnemen voor het verkrijgen van de public domain software.

Verdere informatie van deze werkgroep is te vinden in *bijlage J* van dit verslag.

Werkgroep 8: Nederlandse TEX gebruikersdag

De voorzitter deelt mede dat aan de organisatoren van de TEX dagen in 1989 te Utrecht, onlangs een cadeau is gegeven.

Betreffende de komende TEX-SGML dagen op 31 augustus a.s. te Groningen deelt hij mede dat er reeds het nodige gedaan is: de data zijn vastgesteld in december 1989, call for papers is vroegtijdig uitgegaan, zaalruimtes zijn gereserveerd, er is overlegd met de kantine dienst, een draaiboek is gemaakt en met betrokkenen besproken, begrotingen (voor de dag en de diverse cursussen) zijn gemaakt en terugkoppeld met het SGML en NTG bestuur, informatie over de dag zelf, en over de TEX cursussen, is verspreid o.a . als bijlagen bij het laatste NTG verslag, op TEX-NL, TEXHaX, TEXMaG, ukTEX en aangeboden aan TUGboat, GUTenberg cahiers, DANTE TEXnische komödie, en TEXline. Er is getracht om NIIO subsidie voor de deelnemers te verkrijgen, maar dit is echter niet verkregen. Leveranciers/standhouders zijn benaderd voor deelname. Toezeggingen zijn nog niet allemaal binnen.
De respons voor de bijeenkomst is echter nog laag. Dit geldt zowel voor wat betreft aanmeldingen van presentaties, als ook voor het aantal inschrijvingen voor de dag en de cursussen. De aanwezigen worden dringend verzocht om suggesties door te geven m.b.t. potentiële sprekers, en vooral niet te vergeten zich aan te melden.

Maasdam (Intermedia) deelt namens de SGML Users' Group Holland mede dat er meer gedaan moet worden aan de PR ook voor mogelijke belangstellenden buiten de SGML Users' Group. SGML cursussen en bijbehorende docenten zijn vastgesteld. Folders zullen binnenkort beschikbaar zijn. De plaats van Bleeker (Elsevier) in de commissie wordt overgenomen door van Wijnen (Wolters Kluwer). De voorzitter reageert hierop dat de informatieverspreiding binnen de SGML Holland Users' groep nog niet heeft plaatsgevonden.
Vervolgens verzoekt Maasdam de voorzitter om op korte termijn een datum voor een bijeenkomst van de werkgroep vast te stellen, ten einde de definitieve vorm van de opleidingen, folder, het programma, en het verspreidingsgebied van de bijeenkomstinformatie vast te leggen. Tevens wil hij een mogelijke andere locatie voor de augustus bijeenkomst en cursussen op de voorgestelde werkgroepsbijeenkomst ter sprake brengen, daar hij begrepen heeft dat de locatie Groningen mogelijk problemen zal opleveren.

De voorzitter, uit hoofde van zijn commissiewerk, heeft geen bezwaar tegen een bijeenkomst, alhoewel het uitwerkingen betreft van het eerder genoemde draaiboek, die door genoemden gedaan kunnen worden. Hij voegt er verder aan toe dat hij de komende veertien dagen in Frankrijk zal vertoeven. De sluitingstijd voor inzendingen van lezingen was 1 mei; helaas zijn er weinig lezingen aangemeld. Het werk wordt verder bemoeilijkt doordat een aantal benaderde buitenlandse sprekers nog geen 'ja' hebben gezegd. Naar zijn overtuiging zijn er 'sprekers' aanwezig op deze vergadering. Volgens het draaiboek zouden pas nu de beslissingen over het definitieve programma genomen kunnen worden: wie zijn de sprekers en in welke volgorde in het programma etc. Daarnaast zou pas in juni vastgesteld worden welke cursussen daadwerkelijk door zouden gaan. Het probleem hierbij is echter dat nog niet beide achterbannen geïnformeerd zijn. Dit impliceert naar zijn mening dat die beslissing, helaas, opgeschort zal moet worden.
De voorzitter deelt verder mede dat de locatie RUG zakelijk beschouwd niet in het geding is. Als externe groep (SGML & TEX Users Group) is er zaalruimte gereserveerd, door de voorzitter uit hoofde van de commissie en niet uit hoofde van zijn medewerkerschap bij de RUG. Afgesproken is met RC-RUG dat in juni, afhankelijk van de belangstelling van de cursussen, definitieve reserveringen gedaan zullen worden.
M.b.t. de PR zaken antwoordt de voorzitter dat, naar zijn mening, de hoofd doelgroep vooral de eigen (SGML en TEX) gebruikersgroepen zijn; daar dient allereerst de informatievoorziening plaats te vinden. Natuurlijk moet de informatie ook naar derden gestuurd worden: een persbericht, annonces in diverse bladen e.d.

Maasdam herhaalt zijn standpunt dat er nu op korte termijn eerst een folder gereed gemaakt en verspreid moet worden. Van zijn kant is op dit moment de organisatie zodanig dat alle informatie beschikbaar is. Hij informeert naar de stand van zaken m.b.t. de TEX organisatie.

De voorzitter zegt dat alleen de TEX sprekers nog niet rond zijn, o.a. vanwege de sluitingsdatum van 1 mei en dat geïnviteerde buitenlandse sprekers nog niet toegezegd hebben. Voor de overige informatie zie bijlagen K1 en K2. Hij zegt dat hij denkt in één van de pauzes wel de namen (en titels onder voorbehoud) van (TEX) sprekers ter plekke te kunnen vaststellen en door te kunnen geven aan Jan Maasdam, t.b.v. de folder en nadere PR activiteiten. De aankondiging van de TEX cursussen waren al lang klaar en uitgegaan. Snelle informatieverstrekking t.a.v. de SGML cursussen is belangrijk om in

juni reeds een beslissing te kunnen nemen welke cursussen al dan niet doorgaan.

Van Wijnen (Wolters Kluwer, secretaris SGML) stelt voor om in de komende week één keer bij elkaar te komen om alles kort te sluiten en tot actie over te gaan. De voorzitter herhaalt dat hij veertien dagen in Frankrijk zal zijn en dus niet bij de voorgestelde bijeenkomst kan zijn. Hij heeft er echter het volste vertrouwen in dat als hij straks de namen doorgeeft (sommige buitenlandse sprekers onder voorbehoud) er dan een folder door Maasdam en van Wijnen opgesteld en verstuurd zal kunnen worden. De voorzitter betreurt het dat van het vroegtijdig opgestelde draaiboek wordt afgeweken en er nu zo ad hoc gehandeld wordt.

Met de uitspraak van één van de aanwezigen, dat de voorbereiding van de gemeenschappelijke SGML & TEX dag niet tijdens deze vergadering moet plaatsvinden, wordt de enigszins emotionele discussie gesloten.[1]

Werkgroep 9: Integratie beelden en TEX

Er was niets te melden.

Werkgroep 10: SGML-TEX relatie

Aan het verslag van de werkgroep (*bijlage O*, verslag 23 november 1989) was niets toe te voegen.

Werkgroep 12: Beheerdershandleiding/documentatie

Er was niets te melden.

Werkgroep 13: Nederlandstalige TEX

Eijkhout (KUN) deelt mede dat D. van Leeuwen zich bij de werkgroep heeft aangesloten.
Eind april is er vergaderd. De notulen zijn echter nog niet vrijgegeven. Genoemd worden de bijdragen die uit deze werkgroep voortgekomen en gepubliceerd zijn, dan wel worden, in TUGboat:

- TUGBOAT 10.3: Nederlandse stijlen
- TUGBOAT 11.1: A4
- TUGBOAT 11.3: Babel

De werkgroep zou zich nu kunnen opheffen, doch het is te verwachten dat onderhoud van de software een belangrijke taak zal worden. De stijlen artikel en rapport zijn populair onder de Nederlandse gebruikers, doch een aantal lastige bugs moeten nog opgelost worden.
Eijkhout (KUN) meldt dat hij geen coördinator meer kan zijn, vanwege zijn toekomstige baan in het buitenland.

De voorzitter verzoekt om een generieke stijl uit de veelheid van de nu aanwezige stijlen. Daarnaast zou het aantal stijlen bevroren moeten worden en de flexibiliteit binnen het TEX niveau te leggen. Geantwoord wordt hierop dat LATEX niet gemakkelijk hiervoor is om een generieke stijl te kunnen gebruiken. De interface tussen de LATEX stijl en de kernel veroorzaakt de nodige problemen. Mogelijk moet het geheel hiervoor herschreven worden. LATEX 3.0 zou misschien oplossingen kunnen aanreiken.

Eijkhout deelt mede dat hij vanaf rond de maand augustus/september werkzaam is in Amerika.

Verdere informatie van deze werkgroep is te vinden in *bijlage L* van dit verslag.

Werkgroep 14: Communicatie

Braams (PTT) meldt dat de werkgroep in april bij elkaar is geweest. Discussie heeft toen ondermeer plaats gevonden of bestanden op meer dan één plaats opgeslagen moeten worden. In Nederland zijn er nu twee TEX fileservers, t.w. één in Nijmegen, en één in Utrecht. De eerstgenoemde bevat alleen TEX zaken en wordt beheerd door werkgroep 13 van de NTG. De andere fileserver behoort tot een vakgroep van de RUU. Het is open voor alle netwerkgebruikers. Het TEX gedeelte (schaduwarchief van de Clarkson archive server) wordt onderhouden door van Oostrum (RUU). Totale capaciteit is 600 MByte.

Bij de laatste werkgroepvergadering is tevens gesproken over de ASCII/EBCDIC problemen bij filetransport van server naar gebruiker. Er zijn sterke vermoedens aanwezig dat bij de MC-VAX gateway te Amsterdam bepaalde (2 á 4) karakters verminkt worden. Er is reeds contact met de locale beheerder opgenomen. De problemen treden op als informatie van een UNIX systeem naar een niet UNIX systeem wordt verstuurd (of vica versa). Mogelijk dat op niet al te lange termijn dit probleem opgelost is.

Van der Laan (RUG) stelde voor bovengenoemde problematiek op papier te zetten.

Kuiken vraagt of in principe alles wat aangeboden wordt, op de NTG fileserver te Nijmegen wordt geplaatst. Braams antwoordt hierop dat de werkgroep liberaal is en tot heden nog altijd aan de verzoeken heeft voldaan, zeker als er nog voldoende plaats is. Een door Kuiken aan één van de leden toegestuurde file blijkt per vergissing niet opgenomen te zijn. Verzocht wordt om opnieuw tot actie over te gaan.
Er konden nog geen mededelingen worden gedaan over het gebruik van de NTG fileserver. Dat er door TEX gebruiker veel van het netwerk, inclusief TEX-NL gebruik wordt gemaakt is duidelijk. Geadcht wordt ook om in het SURFbulletin blad hierover een artikel op te nemen.

Gemeld wordt verder dat Ezendam (MARIN/KNMI) vanwege functieverandering zich uit de werkgroep heeft teruggetrokken. De fileserver kan vanaf heden ook geupdated worden door Karman (KUN)
Er wordt benadrukt om optredende communicatieproblemen direct te melden bij de werkgroep.

[1] In de koffiepauze zijn de namen van TEX sprekers doorgegeven.

Werkgroep 15: TEX 3.0

Een nieuwtje op de vergadering: de oprichting van deze werkgroep met als leden Oostrum (RUU; coördinator), Vens (RUU) en Mulders (KUB).
Oostrum meldt dat hij onlangs TEX 3.0 operationeel heeft gemaakt inclusief de laatste versie van LATEX. Hij wil nu het nieuwe fontselectie systeem van Rainer Schöpf en Frank Mittelbach installeren.

Buiten de UNIX omgeving schijnt iemand in Engeland TEX 3.0 actief te hebben. Ook schijnt SbTEX versie 3.0 beschikbaar te zijn.

Discussie vindt vervolgens plaats over hetgeen men van TEX 3.0 kan verwachten. Kort worden de features uiteengezet. Genoemd worden ook de multilingual problemen (gebruik van multilingual parameter aan begin van paragraaf), problemen bij uitgebreide hyphenations, beschikbaarheid voor verschillende systemen.

De taak van de nieuwe werkgroep is vooral om duidelijkheid in bovenstaande te brengen.

5 Verenigingszaken

De voorzitter deelt mede dat de statuten van de vereniging op de server geplaatst zullen worden.
Op de vraag of de vereniging ook ereleden kent, wordt gezegd dat het mechanisme bestaat, maar dat de NTG noch geen ereleden nog heeft.
Genoemd wordt de fout in de begroting zoals deze is vermeld op blz. 13 van het verslag van de 4^e NTG bijeenkomst: de totale inkomsten/uitgaven dient met $f.$ 3.000,- vermindert te worden (optel fout).
N.a.v. de vraag wat de zin in Bijlage D van het laatste verslag, toelichting punt 3: 'dit is gerealiseerd' betekent, antwoordt de penningmeester dat genoemd bedrag het totale netto resultaat is, welke achteraf hoog genoemd mag worden. Dat het bedrag voor de NTG-SGML dag in 1990 veel lager ligt, komt vanwege de gemaakte voorzichtige schatting.
De voorzitter deelt mede dat het aantal aangemelde leden zeer mee valt: 90 contributie betalende leden waaronder 15 instituutsleden. Door de onverwacht hogere post van inkomsten kunnen TEX activteiten behalve mentaal nu ook financieel ondersteund worden.
Op de vraag aan de aanwezigen of het bestuur het vertrouwen heeft om de extra gelden naar eer en geweten zo goed mogelijk te gebruiken, wordt positief mee ingestemd. De penningmeester Braams wordt bedankt voor zijn financieel verslag.

6 Rondvraag

Betreffende de komende SGML-TEX bijeenkomst (31 augustus 1990) komt het volgende ter sprake:

- Naar aanleiding van een vraag waarom er geen TEX beginnerscursus is, wordt geantwoord dat er een keuze gemaakt moest worden. Doel is om zoveel mogelijk cursussen 'vol' te krijgen. Dit jaar is eerst gekozen voor een LATEX introductiecursus. Toch bleek uit de discussie dat er wel degelijk behoefte was aan een TEX beginnerscursus.
- Een cursus in de trent van 'alles wat u wilt weten, doch nooit durft te vragen' zou een goede suggestie voor de bijeenkomst in 1991 zijn.
- Een lezing over het onderwerp 'wat is TEX', is voor 1991 ook een mogelijkheid. Materiaal is bij Poppelier (Elsevier) en Eijkhout (KUN) aanwezig.
- Informatie over de dagen is ook gestuurd naar de redactie van de rekencentra mededelingen.
- Van Gent (KUB) noemt posters als een efficient medium t.b.v. PR van de TEX-dagen. Hij kent iemand die goedkoop het geheel kan ontwerpen en drukken.

Verder komt nog het volgende naar voren:

- De Jong is op zoek naar documenten waarmee hij in zeer korte tijd TEX en Metafont in de praktijk kan brengen.
- Van Oostrum (RUU) heeft twee artikelen over TEX voor het blad PROOF geschreven.

7 NTG Presentaties

Twee presentaties werden deze dag gegeven t.w.:

- 'SGML en TEX in een wetenschappelijke uitgeverij' door Poppelier (Elsevier Science Publishers, PSED, R&D-afdeling),
- 'Communicatie allerlei' door Braams (PTT Dr Neher Laboratorium).

De lezing van Poppelier, welke tevens tijdens Cork'90 gegeven zal worden, is als *bijlage BB* bij dit verslag opgenomen. De belangrijkste punten uit de lezing van Braams waren reeds in het vorige verslag opgenomen.

8 Sluiting

De datum voor de volgende vergadering is op:

dinsdag 20 november 1990

bij DIGITAL te Utrecht; gastheer Theo de Klerk.
De gastheren worden bedankt voor de geboden diensten.
De aanwezigen worden bedankt voor hun bijdragen in de discussie.

De bijeenkomst wordt uiteindelijk om 17:15 uur gesloten.

Getekend:
Voorzitter: Secretaris:

BIJLAGE A

Besluitenlijst

De volgende besluiten zijn van kracht:

4.1 Leden van zuster (TUG)verenigingen betalen voor de NTG-dagen dezelfde toegangsprijs als NTG leden.
Per buitenlandse vereniging wordt één bestuurslid uitgenodigd (betaalt daarbij geen deelnemerskosten).

4.2 In de kascommissie voor 1990 hebben zitting Biegstraaten (TUD) en Evers (RUU).

BIJLAGE B

TEX kalender

10–12	okt	Deutsche TEX (9^e)/DANTE	Göttingen, Duisland
17	okt	ukTEX ug AGM	Aston, UK
4	dec	GUTenberg Tutorial: TEX en fonts	Parijs, Frankrijk
20–22	feb	Deutsche TEX (10^e)	Wenen, Oostenrijk
?	mei	NTG (7^e)	?
28–30	mei	GUTenberg'91	Parijs, Frankrijk
?	aug	TUG	Providence, Rhode Island, USA
23-26	sep	EuroTEX	Parijs, Frankrijk

BIJLAGE C

Werkgroepen Nederlandstalige TEX Gebruikersgroep

1. **Cursusmateriaal (algemeen)**
   A.W.W.M. Biegstraaten (TUD)
   G. Haayer (Styx Publications)
   C.G. van der Laan (RUG) (coördinator)
   J.R. Luyten (RUG)
   P. Tutelaers (TUE)

2. **Richtlijnen voor publicaties/conferentie proceedings etc.**
   T.A. Jurriens (RUG; Sterrenkunde) (coördinator)
   N.A.F.M. Poppelier (TEXnique)
   R. Smedinga (RUG; Informatica)
   J. Soutberg (Elsevier Science Publisher)

3. **Evaluatie produkten (Ned. LATEX incl sty. files en afbreekregels; andere macrocollecties AMSTEX; converters K-talk; TEX naar ASCII; index programmatuur; dBase-TEX koppeling; adreslabels; verkrijgbaarheid etcetc.)**
   J.L. Braams (PTT Research Neher Lab) (coördinator)
   M.A.J.H. Broeren (Océ Nederland B.V.)
   G. Haayer (Styx Publications)
   J.R. Luyten (RUG)
   H.P.A. Mulders (KUB)

4. **Fonts (gebruik van Metafont)**
   H. Brouwer (EGD)
   G. Haayer (Styx Publications) (coördinator)
   A.J. de Meyer (RUU; Wiskunde)
   P. Tutelaers (TUE)
   F.J. Velthuis (RUG; Rekencentrum)

5. **Drivers, previewers, printers, postscript**
   J.L. Braams (PTT Research Neher Lab)
   H. Brouwer (EGD)
   P. Tutelaers (TUE)

6. **Lijst en link met fotozetters**
   G. Haayer (Styx Publications)
   M. Helmig (Philips)
   T.A. Jurriens (RUG; Sterrenkunde)
   F.J. Velthuis (RUG; Rekencentrum) (coördinator)

7. **PC-perikelen; campuslicentie etc.**
   E. Algera (EGD; Amiga)
   G.J. Braas (EGD; Archimedes)
   H. Brouwer (EGD)
   J.R. Luyten (RUG; DOS)
   P. Tutelaers (TUE)
   E.J. Vens (RUG; DOS)
   J.J. Winnink (-; DOS)
   E.B.J. van der Zalm (RUU; Atari)
   R. Veldhuyzen van Zanten (SARA; McIntosh)

8. **Nederlandse TEX gebruikersdag (zomer 1990)**
   C.G. van der Laan (RUG) (coördinator)
   T.A. Jurriens (RUG; Sterrenkunde)

9. **Integratie beelden en TEX**
   H. Brouwer (EGD)
   G.D. Draaijer (MARIN)
   M. Helmig (Philips)
   T.A. Jurriens (RUG; Sterrenkunde) (coördinator)

10. **SGML-TEX relatie**
    A.W.W.M. Biegstraaten (TUD)
    D.C. Coleman (Elsevier Science Publisher)
    C.G. van der Laan (RUG) (coördinator)
    J. Grootenhuis (CIRCE)
    N.A.F.M. Poppelier (TEXnique)

11. (werkgroep is vervallen)

12. **Beheerders handleiding/documentatie**
    J.L. Braams (PTT Research Neher Lab)
    E.J. Evers (RUU; Geneeskunde) (coördinator)

13. **Nederlandstalige TEX**
    J.L. Braams (PTT Research Neher Lab)
    V. Eijkhout (KUN)
    D. van Leeuwen (RUL)
    N.A.F.M. Poppelier(TEXnique)

14. **Communicatie**
    J.L. Braams (PTT Research Neher Lab) (coördinator)
    V. Eijkhout (KUN)
    E.J. Evers (RUU; Geneeskunde)
    P. van Oostrum (RUU)

15. **TEX 3.0**
    H.P.A. Mulders (KUB)
    P. van Oostrum (RUU) (coördinator)
    E.J. Vens (RUG)

BIJLAGE D

Ontvangen Local Guides en andere TEX documenten

nr.	Bedrijf	Auteur	Uitgave	pag	Titel
1.	TRW	Michael Urban	feb-86	56	An Introduction to LATEX
2.	TUE	Michael Urban	feb-86	49	An Introduction to LATEX (Nederlandstalige versie)
3.	TRW	Michael Urban	aug-87	33	A Guide to TEX for theTroff User
4.	TUE	Richard Furuta	sep-87	12	Using LATEX on Berkeley
5.	RUG	R. de Bruin, e.a. [1]	jan-88	188	Publiceren met LATEX
6.	KUB	Rekencentrum	feb-88	27	LATEX-handleiding
7.	MARIN	G.D. Draaijer	aug-88	4	Handleiding SliTEX
8.	RUG	C.G. van der Laan e.a. [2]	sep-88	50	Evaluation of K-Talk
9.	TUD	Hulsen en vd Zanden	dec-88	22	Gebruik van LATEX in de Vakgroep Stromingsleer
10.	TUE	Piet Tutelaers	feb-89	9	Het installeren van PCTEX
11.	DIGITAL	Theo de Klerk	mar-89	40	Using LATEX at SWAS IJSAPL system
12.	RUG	Bert Vunderink	mar-89	36	Local guide to TEX and LATEX on the VAX8650
13.	KUB	Huub Mulders	mar-89	31	KUB/TEX-handleidingen (styles, drivers en previewers)
14.	ACCU	Ad Emmen e.a. [3]	mei-89	39	Tekstproductie met LATEX
15.	RUG	J.B. Hemel e.a. [4]	jun-89	39	Het maken van transparanten voor overhead-projectie
16.	RUG	C.G. van der Laan e.a. [5]	jun-89	31	SGML-LATEX (1. Wiskundige formules)
17.	TUE	Piet Tutelaers	okt-89	87	Sheets van de LATEX cursus
18.	RUG	L. Steemers e.a. [6]	nov-89	54	Journal Style Guidelines
19.	RUG	C.G. van der Laan e.a. [7]	nov-89	50	LATEX Templates; Letter and Article
20.	MARIN	G.D. Draaijer	nov-89	9	Local Guide LATEX en SliTEX
21.	Manitoba	Michael Doob	jan-90	90	A Gentle Introduction to TEX; a manual for Self-study

Niet vermelde TEX/LATEX documenten worden met belangstelling door het bestuur tegemoet gezien.

Aanvragen van NTG leden voor bovengenoemde documentatie dient in eerste instantie rechtstreeks via de auteur c.q. instelling plaats te vinden.

[1] R. de Bruin, C.G. van der Laan, J.R. Luyten, H.F. Vogt
[2] C.G. van der Laan, J.R. Luyten (RC-rapport 22)
[3] Ad Emmen, Evert Jan Evers, Ton Krijnen, André de Meijer, Henk Penning, Nico Poppelier, Hetty Zweerus
[4] J.B. Hemel, C.G. van der Laan, R.C.G.M. Lauwerier (RC-rapport 23)
[5] C.G. van der Laan, D.C. Coleman, J.R. Luyten (RC-rapport 24)
[6] L. Steemers, C.G. van der Laan (RC-rapport 26)
[7] C.G. van der Laan, J.R. Luyten (RC-rapport 27)

BIJLAGE E

TeX-NL subscription

8 october 1990

TeX-NL is de Nederlandstalige TEX-informatie distributielijst (ook wel discussielijst genoemd). Het adres is:

TEX-NL@HEARN

Men kan zich op deze TeX-NL discussielijst abonneren (TEX-NL mails ontvangen en versturen) via de volgende VAX/VMS commando's (of analoge commando's voor andere computer systemen):

```
$
$ SEND LISTSERV@HEARN
  > SUBSCRIBE TEX-NL  your_name
$
```

Een lijst van deelnemers is te verkrijgen via de commando's:

```
$
$ SEND LISTSERV@HEARN
  > REVIEW TEX-NL
$
```

Met als resultaat:

```
*
*  TEX-NL
*
*  Review=       Public
*  Subscription= Open
*  Send=         Public
*  Notify=       Yes
*  Reply-to=     List,Ignore
*  Files=        Yes
*  Validate=     Store only
*  Errors-To=    Owners
*  X-Tags=       Comment
*  Stats=        None,Private
*  Confidential= No
*
*  owner=  U070007@HNYKUN11  (Nico Cox)
*
*
pfuetz@AGD.FHG.DE                       Matthias Pfuetzner, ZGDV Darmstadt
STOKHOF@ALF.LET.UVA.NL                  Martin Stokhof
Z3000PA@AWITUW01                        Hubert Partl
mattes@AZU.INFORMATIK.UNI-STUTTGART.DE  Eberhard Mattes
henk@BEBUX.UUCP                         Henk Dijkstra
GEOMETRY@BGERUG51                       FRANK DE CLERCK
LAAAA18@BLEKUL11                        Erik van Eynde
hans@CS.KUN.NL                          Hans Meijer
phons@CS.KUN.NL                         Phons Bloemen
piet@CS.RUU.NL                          Piet van Oostrum
vansoest@CS.UTWENTE.NL                  Dick van Soest
eijkhout@CSRD.UIUC.EDU                  Victor Eijkhout
GPARTOS@DGIHRZ01                        G. Partosch, HRZ Univ. Giessen, F.R.G.
GPTEX@DGIHRZ01                          TeX-Install., HRZ Univ. Giessen, F.R.G.
DANTE@DHDURZ1                           DANTE e.V.
KNAPPEN@DMZNAT51                        "J"ORG KNAPPEN"
nust@DUTENTB.UUCP                       Jan H Nusteling
abi@DUTIAA.TUDELFT.NL                   Ton Biegstraaten
wiorst5@DUTIWS.TUDELFT.NL               Bert van Zomeren
wbahhul@DUTREX.TUDELFT.NL               Martien Hulsen
wbtrvos@DUTREX.TUDELFT.NL               Ron v. Ostayen
```

kees@DUTTWTA.TUDELFT.NL	C.L. Koster
robk@DUTTWTA.TUDELFT.NL	Rob Kuyper
KROPVELD@HASAMC51	Daniel Kropveld
A410SAKE@HASARA11	Sake J. Hogeveen
A471HANS@HASARA11	hans van der meer
SOND0016@HASARA11	R Veldhuyzen van Zanten
EMMEN@HASARA5	Ad Emmen
dee%svcentlv@HDEDH1	Dick Dee
TNEOCMS@HDETUD1	Folbert Pijper
WBAHKUI@HDETUD1	Gerard Kuiken
WBAHVDZ@HDETUD1	jaap van der zanden
WIORA03@HDETUD1	Netty Zuidervaart
JIAN@HDETUD11	Jian LUO
DENHAAN@HDETUD5	Jack den Haan
WBWEAHA@HDETUD51	J.B.W. HOEBEEK
RCRONH@HEITUE5	Ron Helwig
RCGBBATG@HEITUE51	"Tonnie Geraets
ALDHAHIR@HENUT5	Alaaddin Al-Dhahir
GRAGERTP@HENUT5	Peter Gragert
TWPOLDER@HENUT5	Jan Willem Polderman
henk@HGATENL.HOBBY.NL	Henk Dijkstra
CGL@HGRRUG5	CG VAN DER LAAN
DRUNEN@HGRRUG5	Rudi van Drunen
EGDNT01H@HGRRUG5	Henk Brouwer
TAJ@HGRRUG5	Theo Jurriens
VENS@HGRRUG5	Erik Jan Vens
LEEGTE@HGRRUG51	Henk Leegte
KANABY@HHEOUH51	Abdy Jooya
PUXHHO@HHEOUH51	Bert Houben
KANTGS@HHEOUH52	Ad Gerards
CRISNB@HLERUL2	Nicoline den Breems
LETTVA@HLERUL2	Andrea de Leeuw van Weenen
HERMANS@HLERUL5	FRANK HERMANS
FTHKOPER@HLERUL52	GER KOPER
BORSBOOM@HLERUL53	G.J.J.M. Borsboom
HARTEVEL@HLERUL53	JAN VANHARTEVELT
BUUREN@HLERUL55	Stef van Buuren
DAVID@HLERUL59	David van Leeuwen
BRAAMS@HLSDNL5	Johannes Braams
KOUIJZER@HLSDNL5	Guus Kouijzer
HVMYOPEB@HMARL5	PETER BOVENDEERD
MFAGKCHR@HMARL5	CHRIS EVELO
U070007@HNYKUN11	NIEK COX
U070040@HNYKUN11	Patrick Wever
U070070@HNYKUN11	WILLEM JAN KARMAN
U216002@HNYKUN11	Paul Wackers
U250005@HNYKUN11	Peter-Arno Coppen
U251006@HNYKUN11	Hans Stoks
U253002@HNYKUN11	Constant Cuypers
U279102@HNYKUN11	Theo van den Heuvel
U439019@HNYKUN11	Ton de Haan
U605005@HNYKUN11	Willem Jan Karman
U605008@HNYKUN11	Rik Fleuren
U641012@HNYKUN11	Rini van Doorn
MBFYS_NU@HNYKUN51	"Hong Zhou"
BISON@HNYKUN53	"Pieter Bison"
SYLVIA@HNYMPI51	"Sylvia Aal"
GAVIN@HNYMPI52	"Gavin Burnage"
ESU0115@HPEENR51	Rob Wervelman
ESU0201@HPEENR51	Huijsmans R.H.M.
ESU0203@HPEENR51	Gerard D. Draaijer
ESU0507@HPEENR51	Jos Winnink
VANNES@HPEENR51	Gerard van Nes
BART@HROEUR5	bart de vries
VDENDE@HROEUR5	"Jan van der Ende"
HUYGEN@HROEUR51	PAUL HUYGEN
GENT@HTIKUB5	"Joop van Gent"
LEIDELME@HTIKUB5	K.C. Leidelmeyer
S172HMUL@HTIKUB5	Huub Mulders
EVERS@HUTRUU53	Evert Jan Evers / Rijksuniv. Utrecht
POPPELIE@HUTRUU53	"Nico Poppelier"
HOOFT@HUTRUU54	Rob Hooft
HAAN@IRIMCF.TUDELFT.NL	Henk de Haan
SURF114@KUB.NL	Frank Poppe

```
WATSON@LCC.EDU                          Claude M. Watson
andre@MAESTRO.HTSA.AHA.NL               Andre v.d. Vlies
demeijer@MATH.RUU.NL                    Andre de Meijer
soos@MATH.UTWENTE.NL                    Adwin Soos
aerts@MEDIA01.UUCP                      Ad H. Aerts
MBR@OCE.NL                              Marius Broeren
hd@PHILTIS.CFT.PHILIPS.NL               Henk Davids
M_Kooij@PTTRNL.NL                       Martin Kooy @ PTT Research Neher Laborat
PZF5HZ@RUIPC1E                          Frank Mittelbach
BILT@RUUMTC.TCU.RUU.NL                  "A. van der Bilt"
nspit@RUUNSA.FYS.RUU.NL                 "Werenfried Spit"
RWBEST@SARA.NL                          Robert W. Best
jaap@SCI.KUN.NL                         Jaap Bril
petervc@SCI.KUN.NL                      Peter van Campen
NETNEWS@TREARN                          NETNEWS SERVER
STELP@TUDGV1.TUDELFT.NL                 David Stelpstra
HUISMAN@TUDW03.TUDELFT.NL               H. Huisman
rcpt@URC.TUE.NL                         Piet Tutelaers
*
* Total number of "concealed" subscribers:         2
* Total number of users subscribed to the list:  113  (non-"concealed" only)
* Total number of local node users on the list:    0  (non-"concealed" only)
*
```

Opmerkingen:

- Verzocht wordt om de TEX-NL listserver niet te gebruiken voor het versturen van grote bestanden (programma's) indien van het alternatief: **de TEX-NL fileserver** (*zie bijlage F*), gebruik gemaakt kan worden.
- Daar ook enkele buitenlanders meeluisteren, wordt men verzocht de 'subject' van de mail in het Engels op te geven.
- De TEX-NL listserver is bij uitstek geschikt voor een verzoek voor ondersteuning bij een TEX/LATEX/driver probleem, voor vragen over beschikbaarheid van bepaalde software modulen, voor aankondigingen van bijeenkomsten en/of cursussen, voor het attenderen op bepaalde publicaties, voor het attenderen op bepaalde produkten, voor een mededeling die ook voor een grotere groep interessant is, etcetc..

BIJLAGE F

NTG fileserver faciliteiten

8 october 1990

Sinds mei 1989 heeft NTG de TEX-NL fileserver. Voor leden interessante files worden daarbij centraal beschikbaar gesteld.
Men kan files van deze fileserver betrekken via de volgende VAX/VMS commando's (of analoge commando's voor andere computer systemen):

```
$
$ SEND LISTSERV@HEARN
  > GET filename filetype
$
```

Waarbij de mogelijke *filenames* en *filetypes* in de hieronder getoonde listing zijn opgenomen.

De lijst van alle aanwezige files is te verkrijgen via de commando's:

```
$
$ SEND LISTSERV@HEARN
  > GET TEX-NL FILELIST
$
```

Met als resultaat:

```
*  TEX-NL FILELIST for LISTSERV@HEARN.
*
*  TeX-NL Filelist
*
*  Contains
*      -- general TeX stuff (implementations for micros, graphical
*         shells, printer drivers)
*      -- specifically Dutch stuff (styles and options, hyphenation
*         patterns)
*      -- Dutch TeX Users Group (NTG) stuff
*
*  ****************************************************************
*
*  This file lists the programs that are stored on LISTSERV and can be
*  retrieved by network users.
*
*  If an entry shows nrecs=0 the file is not available.
*
*  This filelist may be sorted in columns 47 to 63 to get a list of
*  files in the order of their updates. Sorting in descending order
*  shows the most recently updated files at the top.
*
* ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
*
*  The GET/PUT authorization codes shown with each file entry describe
*  who is authorized to GET or PUT the file:
*
*     ALL = Everybody
*     N/A = Not Applicable
*     LCL = Local users, as defined at installation time
*     PRV = Private, ie list members
*     OWN = List owners
*     NAD = Node Administrators, ie official BITNET/EARN contacts
*     CTL = LISTSERV Controllers (Also called "Postmasters")
*
*:    NTG = 'BRAAMS@HLSDNL5',   /*  Johannes Braams                  */
*:          'BRAAMS@HLSDNL50',  /* Johannes Braams                   */
*:          'BRAAMS@HLSDNL51',  /* Johannes Braams                   */
*:          'U641000@HNYKUN11', /* Victor Eijkhout                   */
*:          'U641001@HNYKUN11', /* Victor Eijkhout                   */
*:          'EVERS@HUTRUU53'    /* Evert Jan Evers                   */
*
* ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
**************************************************************************
*
*  Dutch hyphenation patterns
*
*  Hyphen1 TeX : short but satisfactory (author: Peter Vanroose)
*  Hyphen1 DOC : some notes by Peter Vanroose
*  Hyphen2 TeX : long and powerful (author: Celex, Nijmegen)
*                Note that this requires stretching the 'triesize'
*                of both TeX and IniTeX!
*  UShyphen ADD: extra patterns to handle the Tugboat exception log
*                (author: Gerard Kuiken)
*
**************************************************************************
*                               rec               last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  HYPHEN1  TEX        ALL NTG V      73   335 89/05/19 20:19:19
  HYPHEN1  DOC        ALL NTG V      78    34 89/06/26 12:26:25
  HYPHEN2  TEX        ALL NTG V      72  7949 89/05/19 20:21:37
  USHYPHEN ADD        ALL NTG V      73   378 90/05/14 13:20:11

**************************************************************************
*
*  Options for Dutch
*
*  A4 STY    : A4-paper width and height
*              by Nico Poppelier and Johannes Braams (historical order)
*              Note that this is not the A4 option of John Pavel.
*  A4 TeX and A4 DOC: Accompanying documentation for A4.STY
*  Dutch STY : Redefines captions and does other useful things for
*              all standard document styles. (author: Johannes Braams)
*              This is really an international option.
*  German STY: The style on which 'Dutch' was based. The two are
*              compatible. (author: Hubert Partl)
*  Sober STY : Reduces section headings and white spaces a bit;
*              this is only repair for the standard styles. The official
*              NTG styles (below) can do without. (author: Nico Poppelier)

**************************************************************************
*                               rec               last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  A4       STY        ALL NTG V      71   111 90/03/12 15:44:51
  A4       DOC        ALL NTG V      75   501 90/03/12 15:45:31
  A4       TEX        ALL NTG V      73    19 90/03/12 15:46:01
  DUTCH    STY        ALL NTG V      80   397 89/05/27 17:30:55
  GERMAN   STY        ALL NTG V      76   364 89/06/24 16:08:38
  SOBER    STY        ALL NTG V      77   147 89/06/24 16:06:16

**************************************************************************
*
*  Dutch styles (author: Victor Eijkhout)
*
*  Completely compatible to 'article' and 'report', but improved layout;
*  these styles have as default language English,
*  for Dutch or German add corresponding style options
*
*  Artikel1 STY : Article-compatible, tight look
*  Artikel2 STY : Article-compatible, heavily indented; quite something else
*  Artikel3 STY : Article-compatible; zero parindent, positive parskip;
*                 otherwise similar to Artikel1
*  Rapport1 STY : Report-compatible; looks like Artikel1
*  Rapport2 STY : will probably not come into being.
*  Rapport3 STY : Report-compatible; looks like Artikel3
*  Boek STY     : Book-compatible; artikel1 layout
*
*  Options for the Dutch styles
*
*  Ntg10 STY    : 10point option for all styles
*  Ntg11 STY    : 11point option for all styles
*  Ntg12 STY    : 12point option for all styles
*  Voorwerk STY : Replaces Titlepage.STY for report styles
*
*  NTGstyle UUE : All in one buy; UUencoded ZOO archive (see below
```

```
*                for ZOO)
************************************************************************
*                              rec               last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  ARTIKEL1 STY        ALL NTG V      77   626 90/04/25 11:19:32
  ARTIKEL2 STY        ALL NTG V      77   534 90/04/25 11:19:57
  ARTIKEL3 STY        ALL NTG V      77   631 90/04/25 11:20:21
  RAPPORT1 STY        ALL NTG V      80   642 90/04/25 11:20:59
  RAPPORT2 STY        ALL NTG .       .     0 ........ ........
  RAPPORT3 STY        ALL NTG V      80   641 90/04/25 11:20:43
  BOEK     STY        ALL NTG V      80   679 90/04/25 11:19:12
  NTG10    STY        ALL NTG V      78    59 89/08/03 10:46:18
  NTG11    STY        ALL NTG V      76    60 89/08/03 10:46:46
  NTG12    STY        ALL NTG V      75    58 89/08/03 10:47:10
  VOORWERK STY        ALL NTG V      69    58 90/04/17 12:38:26
  NTGSTYLE UUE        ALL NTG V      62  2381 90/05/07 11:47:12

************************************************************************
*
*  The letter style according to Dutch NEN norms (by Victor Eijkhout)
*
*  BRIEF STY    : The style file
*  BRIEF TeX    : An example letter
*  BRIEFDOC TeX : Explanation of the options of the letter style
*
************************************************************************
*                              rec               last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  BRIEF    STY        ALL NTG V      79   644 90/02/07 12:24:57
  BRIEF    TEX        ALL NTG V      72   198 89/10/24 16:48:49
  BRIEFDOC TEX        ALL NTG V      69   278 89/11/20 09:34:53

************************************************************************
*
*  The latest in TeXnology
*
*  LATEXSRC UUE : Latest versions of all LaTeX materials;
*                 UUencoded ZOO archive
*  ASCII TeX    : ASCII table (author: Victor Eijkhout)
*  TUGBOAT COM  : Common commands for Tugboat styles
*  TUGBOAT STY  : Plain TeX style for Tugboat article
*  LTUGBOAT STY : LaTeX style for Tugboat articles
*
*  BTXMAC.TEX   : BibTeX 0.99c macros for use with plain TeX.
*                 The file specifies that is meant for TeX 3.0 or later
*
************************************************************************
*                              rec               last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  LATEXSRC UUE        ALL NTG V      62 10513 90/01/04 10:44:38
  ASCII    TEX        ALL NTG V      82   194 89/10/20 20:06:20
  TUGBOAT  COM        ALL NTG V      80   797 90/08/28 18:44:06
  TUGBOAT  STY        ALL NTG V      80  2278 90/08/28 18:50:07
  LTUGBOAT STY        ALL NTG V      80   444 90/09/07 11:07:53
  BTXMAC   TEX        ALL NTG V      80   624 90/08/15 16:59:21

************************************************************************
*
*  Pleasant reading material about TeX and its uses
*
*  NTGstyle TeX : Manual for the Dutch LaTeX styles
*  Layout TeX   : Article about documentstyle development in LaTeX
*                 Intended as supplement to chapter 5 LaTeX book
*  Layout2 TeX  : Goes with previous; in German (Hubert Partl)
*  Refman STY   : Needed for previous two
*  Bridge TeX   : About setting bridge games in LaTeX (Kees van der Laan)
*  Artdoc TeX   : The history of the 'Artikel' styles; almost a
*                 manual for document style development; in Dutch
*  Rapdoc TeX   : The same for the 'Rapport' styles (Victor Eijkhout)
*  Gentle TeX   : A Gentle Introduction to TeX (Michael Doob)
*
```

```
*************************************************************************
*                              rec              last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  NTGSTYLE TEX        ALL NTG V      72   122 89/09/04 12:20:21
  LAYOUT   TEX        ALL NTG V      79  1090 89/06/26 12:12:34
  LAYOUT2  TEX        ALL NTG V      80  1011 89/06/26 12:03:15
  REFMAN   STY        ALL NTG V      79   492 90/03/26 18:12:17
  BRIDGE   TEX        ALL NTG V      75   415 89/06/26 11:54:36
  ARTDOC   TEX        ALL NTG V      71   708 89/09/04 12:21:36
  RAPDOC   TEX        ALL NTG V      75   735 89/09/04 12:22:13
  GENTLE   TEX        ALL NTG V      79  5341 90/01/09 13:56:01

*************************************************************************
*
*  Nederlandstalige TeX Gebruikersgroep (Dutch TeX Users Group)
*
*  Notuul1 TeX  : Vergadering 23 juni 1988
*  Notuul2 TeX  : Vergadering 24 november 1988
*  Notuul3 TeX  : Vergadering 11 mei 1989 (3 bestanden: notuul3a,b,c)
*  TeXdag89 TeX : Verslag eerste Nederlandse TeXdagen 29/30 juni 1989
*
*************************************************************************
*                              rec              last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  NOTUUL1  TEX        ALL NTG V      80  1043 89/06/26 11:50:54
  NOTUUL2  TEX        ALL NTG V      80  1457 89/06/26 11:52:06
  NOTUUL3A TEX        ALL NTG V      80   108 89/10/30 10:12:47
  NOTUUL3B TEX        ALL NTG V      80  1941 89/10/30 10:13:27
  NOTUUL3C TEX        ALL NTG V      80  2634 89/10/30 10:14:05
  TEXDAG89 TEX        ALL NTG V      73   274 89/12/08 13:04:52

*************************************************************************
*
*  TeX for micros
*
*  STZOO UUE    : UUencoded ARC archive with ZOO for the Atari ST
*  MSZOO UUE    : zoo.exe for MS-DOS, UUencoded
*  MSFIZ UUE    : fiz.exe for MS-DOS, UUencoded
*  MSSUP201 UUE : MS-DOS support for zoo, ZOO archive, UUencoded
*  Z201SRC1 UUE : Sources of zoo, part 1, ZOO archive, UUencoded
*  Z201SRC2 UUE : Sources of zoo, part 2, ZOO archive, UUencoded
*  TEXSHELL UUE : TeX environment for the Atari ST, ZOOed
*  WP2LATEX UUE : WordPerfect to LaTeX translator, ZOOed
*
*************************************************************************
*                              rec              last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  STZOO    UUE        ALL NTG F      61  2181 89/12/14 12:33:31
  MSZOO    UUE        ALL NTG V      61   946 90/03/18 20:53:06
  MSFIZ    UUE        ALL NTG V      61   297 90/03/18 20:54:11
  MSSUP201 UUE        ALL NTG V      61   621 90/03/18 20:55:29
  Z201SRC1 UUE        ALL NTG V      61  1975 90/03/18 20:58:36
  Z201SRC2 UUE        ALL NTG V      61  2046 90/03/18 20:59:21
  TEXSHELL UUE        ALL NTG V      62   594 90/01/03 16:12:11
  WP2LATEX UUE        ALL NTG V      80  1229 90/03/12 15:58:03

*************************************************************************
*
*    METAFONT sources
*
*    AMSREAD.ME   : A few notes about the contents of AMSFONTS.UUE
*    AMSFONTS.UUE : The AMS font collection UUencoded ZOO archive
*                   split in ten pieces of app. 100kByte
*
*************************************************************************
*                              rec              last - change
* filename filetype   GET PUT -fm lrecl nrecs   date     time   File description
* -------- --------   --- --- --- ----- ----- -------- -------- ----------------
  AMSREAD  ME         ALL NTG V      74    45 90/08/02 14:18:33
  AMSFONTS UU1        ALL NTG F      80  1644 90/08/02 15:50:09
  AMSFONTS UU2        ALL NTG F      80  1643 90/08/02 15:58:11
```

```
AMSFONTS UU3        ALL NTG F      80  1643 90/08/02 16:07:46
AMSFONTS UU4        ALL NTG F      80  1643 90/08/02 16:41:15
AMSFONTS UU5        ALL NTG F      80  1643 90/08/02 16:44:37
AMSFONTS UU6        ALL NTG F      80  1643 90/08/02 16:47:16
AMSFONTS UU7        ALL NTG F      80  1643 90/08/02 16:49:31
AMSFONTS UU8        ALL NTG F      80  1643 90/08/02 16:51:56
AMSFONTS UU9        ALL NTG F      80  1643 90/08/02 16:53:58
AMSFONTS UUA        ALL NTG F      80  1659 90/08/02 16:55:44
```

Het adres van de Duitse fileserver (Heidelberg) is:
LISTSERV@DHDURZ1

Voor NTG leden die niet op een netwerk zijn aangesloten, kunnen de meeste files via diskettes verkregen worden. Nadere informatie hierover bij het secretariaat.

BIJLAGE G

TEX stuff at cs.ruu.nl

Revision level: 1.1 - Date: Sep. 20, 1990

Piet van Oostrum,
Dept. of Computer Science,
Utrecht University, The Netherlands
piet@cs.rr.nl

The RUU archive

Here is a brief description of our archive of TeX stuff.

The archive is available by FTP and by mail server. It contains various things, a.o. Atari ST software, GNU software, some Unix software and a lot of TeX things. I will concentrate on the TeX stuff in this message.

How to get the software

By FTP

FTP archive.cs.ruu.nl [131.211.80.5], username anonymous or ftp, and your address (user@host) as password. All user accessible files are in directory pub.
Note that this is a Unix system, so a file in a directory is accessed as `dir/subdir/file`.

By mail server

Send a message to `mail-server@cs.ruu.nl` with any of the commands mentioned at the end of this message. If you use the mail server, the 'pub/' part of file names is not used.

Mailer problems

Sometimes the mail server is unable to reply to a person. We will recognize any internet address with a valid MX record. If you don't get a reply within a few days, try a TEST message with a PATH command. There are a few known reasons why an address is not recognised:

Your mailer inserts a wrong "from" address. E.g. an internet style address while using a non-registered domain. In this case let your management register a proper internet domain. Or a hostname without an MX record. In this case you mignt use a different hostname that knows about you.

UUCP sites might try the following syntax: user@host.uucp or host!user BITNET users may try user@host.bitnet. Some BITNET mailers give user@host as "from" address which is bound to fail.

If you are reachable from a known internet site (e.g. a.b.c), you may try `user%host@a.b.c` For example if you are a UUCP site connected to uunet, the following might work:
`user@host.uucp` or
`uunet.uu.net!host!user` or
`user%host@uunet.uu.net`
Never mix ! and @ style addresses, this is guaranteed to give problems.

A source route address::
`<@uunet.uu.net,@host1,@host2:user@host>`
may also work.

Character set problems

If the message have to go through BITNET to reach you, there may be problems with character translations. Most notably are the curly brackets or braces ({}) that may be translated to character codes above 128 in the ASCII table. Sometimes also carets (^) and tildes (~) will get corrupted. In this case you may try to use uuencoded or btoa'ed transmission (see below).
In the near future I hope to supply also xxencoded transmission that is better resistant againgst network translations.

What is in the archive?

The following subdirectories and files are available:

ATARI-ST	-	A lot of public domain Atari 1040ST software (games, compilers, utilities, ...). Most of it retrieved from the UseNet distributions. See the file ATARI-ST/INDEX. There is also an ATARI-ST/tex subdirectory with tex, dvi drivers and previewers.
DOC	-	Various documentation; among others Internet worm reports, a BSD socket programmer's primer.
ELM-2.3	-	The famous Elm Mail User Agent, version 2.3. The entire package and the PostScript documentation of course.
FTP-LIST	-	a set of files, maintained by Jon Granrose, containing information about anonymous FTP sites worldwide.
GNU	-	Various (up to date) software from the GNU project.

HP-UX	-	Software that will most likely run under HP-UX 6.5 and what's also a good start for other System V alikes. See INDEX file in this directory for more information.
NN-6.4	-	Kim Storm's famous news reader, version 6.4. Contains the official release as well as official patches.
TEX	-	(La)TeX software as well as the TeXhax and TeXmag articles. Also the *entire* TeX 3.0 distribution.
UNIX	-	Various UNIX software, easy-to-run on BSD systems as well as System V systems (with BSD enhancements). See the INDEX file in this directory for more information.
ls-lR.Z	-	A "ls -lR" listing of the entire archive.

Most directories contain a file INDEX with a brief description of the contents.

The TEX subdirectory contains back issues of TeXhax, TeXmag, UKTeX, the TeX3.0 distribution from labrea, a copy of the latexstyle and bibtexstyle archives from clarkson, a number of DVI drivers (this collection will grow in the future), utilities like bibtex and makeindex, the files from Mittelbach and Schöpf (in the latexstyle directory).

A number of directories will be updated automatically each week from the labrea and clarkson archives.

The following listing is the INDEX file in the TEX subdirectory. For information contact Piet van Oostrum <piet@cs.ruu.nl>

```
drwxr-xr-x   6 piet     staff       1024 Sep 20 22:33 AMS
drwxr-xr-x   2 piet     staff       1024 Sep 21 01:16 DVI
drwxr-xr-x   2 piet     staff       1024 Sep 20 22:39 FONTS
-rw-r--r--   1 piet     archive     3924 Sep 14 17:10 INDEX
-rw-r--r--   1 piet     staff      37155 Feb 25  1990 MuTeX_doc.tar.Z
drwxr-xr-x   2 piet     staff       1024 Sep 20 22:34 NTG
drwxr-xr-x  13 piet     staff       1024 Sep 28 15:10 TEX3.0
drwxr-xr-x   2 piet     staff       1024 Sep 20 22:35 bibtex
drwxr-xr-x   2 piet     staff       1024 Sep 20 22:29 bibtexstyle
-rwxr--r--   1 piet     staff      76278 Dec  8  1989 diagram.tex
-rwxr--r--   1 piet     staff      39944 Dec  8  1989 diagramdoc.tex
-rw-r--r--   1 piet     staff      28426 Sep 14 17:01 diagrams.tex
-rw-r--r--   1 piet     staff      55462 Sep 14 17:01 diagrams_doc.tex
-rw-r--r--   1 piet     staff      13018 Apr  5 12:01 edmac.doc
-rw-r--r--   1 piet     staff      34321 Apr  5 12:01 edmac.tex
drwxr-xr-x  14 piet     staff       1024 Sep 20 22:39 emtex
-r--r--r--   1 henkp    archive    30763 Jun 15 12:22 hyphen.dutch.Z
drwxr-xr-x   4 piet     staff       4096 Sep 20 22:30 latexstyle
-rwxr--r--   1 piet     staff     390017 Jun 14 10:45 makeidx.zoo
-rw-r--r--   1 piet     staff      97113 Apr  2 16:51 mtex.tar.Z
-rwxr--r--   1 piet     staff      93579 Dec 13  1989 mtexfonts.tar.Z
-rw-r--r--   1 piet     staff     401659 Jan 22  1990 spiderweb.tar.Z
drwxr-xr-x   7 piet     archive     1024 Sep 20 22:29 texhax
drwxr-xr-x   6 piet     archive     1024 Sep 20 22:29 texmag
drwxr-xr-x   2 piet     staff       2048 Sep 20 22:29 tugboat
drwxr-xr-x   2 piet     staff       2048 Sep 20 22:33 uktex
-rw-r--r--   1 piet     staff      70425 Apr 10 18:08 wp2latex.arc
```

INDEX	-	This file
AMS	-	AMSTeX, AMS-LaTeX and AMSFonts (the new versions)
DVI	-	directory with various dvi convertors
FONTS	-	fonts and utilities for metafont
NTG	-	directory with submissions from NTG (Dutch users group)
TEX3.0	-	The distribution files for TEX3.0
bibtex	-	the bibtex program
bibtexstyle	-	bibtex style files from clarkson
diagram.tex	-	Macros for commutative diagrams (from Francis Borceux)
diagramdoc.tex	-	Documentation for the above
diagrams.tex	-	Another commutative diagram package (from Paul Taylor)
diagrams_doc.tex	-	Documentation for the above
emtex	-	complete TeX/Metafont package for MSDOS
hyphen.dutch.Z	-	Dutch hyphenation patterns (compressed)
latexstyle	-	a copy of the latex-style directory at clarkson.edu
makeidx.zoo	-	The famous makeindex program (sources) version 2.9
mtex.tar.Z	-	tex macros and metafont files for music typesetting, includes English manual
MuTeX_doc.tar.Z	-	Manual only from the above
mtexfonts.tar.Z	-	tfm files and 300dpi PK files for the above (not necessary if you have metafont)
spiderweb.tar.Z	-	a WEB system for other programming languages than Pascal
texhax	-	back copies of texhax magazine (texhax/19xx/yyy) A list of subjects per year is in texhax/19xx/Subjects. Some years have an index in texhax/19xx/INDEX.
texmag	-	back copies of texmag magazine (texmag/19xx/yyy)
tugboat	-	a copy of the tugboat directory at labrea.stanford.edu
wp2latex.arc	-	an MSDOS program to convert Wordperfect 5.0 files to LaTeX (documentation is in Dutch)

The HELP file

How to use the RUU CS Mail Server

The RUU CS Mail Server can be reached at e-mail address "mail-server@cs.ruu.nl". If your mailer doesn't understand domain addresses, try a nearby gateway like "mcsun".

When the Mail Server receives a message, It reads the mail headers to determine the requester's address. If a "Reply-To:" header is found, the indicated address is used. If not, it uses the address as specified in the "From:" header.

The message body is scanned for server commands. Every line in the message should contain a valid server command.

A report is sent to you by return mail. Any requests will be handled as soon as the load of the server system permits.

NOTE: *The mail server is still under development. Things may -and will- change in the near future.*

The Server's Archives

RUU CS maintains a number of archives, which are accessible via the mail server. You may also try to use anonymous FTP to access the archives at "archive.cs.ruu.nl" [131.211.80.5].

Files are stored in the archives in one of the following formats:

- **Plain**
  normal ASCII text.
- **Shell Archive**
  ASCII files which can be unloaded using the Unix sh(1) program.
  Shell Archives have names ending in ".shar".
- **Compressed**
  16-bit compression using the compress(1) utility. Compressed files have names ending in ".Z".
- **Tar**
  Standard UNIX tar(1) format. Tar archives have names ending in ".tar".
- **Compressed Tar**
  Compressed tar archive. Compressed tar archives have names ending in ".tar.Z".
- **Arc**
  Standard "arc" format. These files have names ending in ".arc".
- **Zoo**
  Standard "zoo" format. These files have names ending in ".zoo".

When requesting a file you do not have to specify the format-specific extension. A request for a file "foo" will automatically be changed to "foo", "foo.tar", "foo.shar", "foo.Z", "foo.tar.Z", "foo.arc" or "foo.zoo" whichever is available.

The file "INDEX" contains a list of the top-level directories in the archives.

The file "ls-lR.Z" contains a compressed directory of all the files in the archive. This file is updated daily. The file is in Unix directory format. If you are not familiar with Unix, the following should give you a clue: Entries are of the form:

```
./ATARI-ST:
total 184
-rw-r--r--   2 atari    archive    37201 Apr 25 20:33 INDEX
drwxr-xr-x   2 atari    archive     1024 Mar 14 15:15 doc
```

This means this piece is from the directory ATARI-ST. The first file is the file INDEX, which is 37201 bytes long and made on April 25, 20:33 local time. Most directories have a file INDEX giving some more information about its contents.
The last line starts with a 'd', signifying that it is a directory. The contents of the directory will be specified furtheron in the file. If you want to get e.g. the file bombs.arc from the doc directory you have to ask for ATARI-ST/doc/bombs.arc (i.e. the slash character '/' separates directories and files in a complete file specification).

Command Syntax

A command consists of a keyword (verb), followed by zero or more arguments, depending on the command. Command verbs may be specified in all uppercase letters, lowercase or whatever mixed case. In other words: case is not significant in command verbs. Case *IS* significant in command arguments. Empty lines are ignored.

The following commands are understood by the server:

- **BEGIN**
  Begin processing. All commands and errors before this command are forgotten. Use this if your mailer inserts garbage.
- **PATH <address>**
  The return address used by the server is set to the indicated <address>. This must be a valid address by which you can be reached. It may contain a domain-based address.
  Use this command if you are not sure that the return addresses generated by your mail system are reliable.
- **END or EXIT**
  The remainder of the message is ignored. This can be useful if a .signature is appended to the message.
- **LIMIT <number>**
  Specify the maximum number of bytes which may be sent in a single mail message. Transfers exceeding this amount will be split before sending.
  The amount may be specified in Kbytes, e.g. "30K". The default value is 64K.
  NOTE: setting the limit will only affect "send" and "resend" commands following this command. NOTE: due to mailer overhead, it is possible that the size of the mail which reaches you will (slightly) exceed this limit.
- **UUENCODE**
  The requested items will be uuencoded before sending. Because most archives are binary files, they are always encoded before sending. Uuencoding is default.
  Use "send uudecode" to obtain the uudecode/uuencode programs. NOTE: setting the encoding will only affect "send" and "resend" commands following this command.
- **BTOA**
  The requested files will be encoded using "btoa". Btoa encoded files are smaller than uuencoded files, but not everyone can handle btoa encoded files yet.
  Use "send btoa" to obtain the btoa/atob coding programs. NOTE: setting the encoding will only affect "send" and "resend" commands following this command.
- **SEND <item>**
  The specified <item> is looked up in the server archives. If found, it will be sent to you by e-mail. Multiple items may be specified with one SEND command. If <item> is a directory, a listing of the directory will be sent, rather than the directory itself (but see the description of the PACK command).
  NOTE: the names of the <item>s are case sensitive! On Unix a file within a directory is given as dir/file.
- **RESEND <item> <part> [<part>...]**
  Re-send the indicated <part>s of this item. This is useful if not all parts of a multi-parts transmission did arrive correctly. When re-transmitting, the encoding and limit used must be identical to those of the original transmission.
- **PACK <archive>**
  This command tells the server that directories should be sent as archives. If you give this command, all subsequent SEND commands that specify a directory will cause the directory (together with all its subdirectories) to be sent as an archive. This is only allowed when the directory contains no more than 2MB of files. <archive> must be one of:
  tar - for a compressed tar file
  arc - for an arc file (each subdirectory will be sent as a separate arc file)
  zoo - for a zoo file
- **INDEX**
  This is equivalent to "send INDEX".
- **HELP**
  This command gives a list of server commands.
- **TEST**
  This command is for testing. No files will be sent if you use this, but a confirmation message will be sent to the return path as determined from the mail headers or the "path" command. You may use this to find out if your address is valid, and to check the status of your request.

Sample Mail Server Report

Sending:

```
path jv@mh.nl
btoa
send INDEX bio
resend zoo 2 3 4
send foo
end
```

will generate the following report:

```
        From: RUU CS Mail Server <mserv@cs.ruu.nl>
[1]     To: jv@mh.nl
        Subject: Your request
        Date: Sun, 1 Oct 89 18:25:39 MET (+0100)

        RUU CS Mail Server 1.6

        Processing mail headers ...
[2]     Return address: "Johan Vromans <jv@mhres.mh.nl>"

        Processing message contents...

        Command: path jv@mh.nl
[3]     => Return address: "jv@mh.nl"

        Command: btoa
        => Encoding = btoa

        Command: send INDEX bio
        => Send: INDEX
        => Send: bio

        Command: resend zoo 2 3 4
        => Resend: zoo, parts 2,3,4

        Command: send foo
        => Send: foo

        Command: end
        => Okay

        Your message has been processed.
        The following requests have been queued:

          request    format (uncoded size) encoding limit remarks
          -------    --------------------- -------- ----- -------
[4]       INDEX      Plain (5K)            plain
          bio.tar.Z  Compressed Tar (6K)   btoa       64K
          zoo.tar.Z  Compressed Tar (171K) btoa       64K Parts 2,3,4 only
          foo        Unknown                              Request skipped

        These requests will be sent to you as soon as the load of
        the server system permits.

        Mail Server finished.
```

As you can see, the return mail is sent to the address [1] indicated by the PATH command [3]. If the PATH command were not issued, the address from the message header [2] would have been used. In the list of requests [4] the format and the size of the files are shown. Note that the size is the size *before encoding*! Finally, the request "foo" could not be found and is skipped.

Some time later the following mails will arrive:

```
From                Size      Subject
------------------- --------- --------------------------------
RUU CS Mail Server   73/4301  "INDEX (complete) ascii"
RUU CS Mail Server   96/7394  "bio.tar.Z (complete) btoa encoded"
RUU CS Mail Server  829/65453 "zoo.tar.Z (part 2 of 4) btoa encoded"
RUU CS Mail Server  829/65453 "zoo.tar.Z (part 3 of 4) btoa encoded"
RUU CS Mail Server  325/25578 "zoo.tar.Z (part 4 of 4) btoa encoded"
```

Files which are sent in parts have all pieces clearly marked as such:

```
------ begin of zoo.tar.Z -- btoa encoded -- part 2 of 4 ------
#(_0M#C)R-&3BEIu9#I[oEFn;50r5kb6%CJq%=NMgE3in‘tMpnX0rOEYPWNM...
=69S\PiSodA"*lArTZ.-(g6DL2A6_5>DMuFV/&S7H/]XEgLe(l@e;-Rqr:iZ...
...
...
$‘eP&iGea"a#e[F!oeo1r@U/FP;::i"V)j_EW+.(U*&IrTJ+u’9=$MY7s*CC...
uI=a5*Wj^#1LD,&>MZKY@H1_a9QE$$4[+?[ePhh"h2Ub"/a,(ES*ZH"nK"6d...
------ end of zoo.tar.Z -- btoa encoded -- part 2 of 4 ------
```

Unpacking

You may obtain the following packages from the server:

btoa	btoa/atob support programs
uudecode	uuencode/uudecode support programs
compress	compress/uncompress support programs

These packages are send unencoded, in "shar" format.

If you have Larry Wall’s "perl" program, you can retrieve the program "unpack.pl" from the server. This will help you in unpacking the mails.
By the way: the server software is also written in perl.

For questions, information and remarks:

Piet van Oostrum [piet@cs.ruu.nl]
Edwin Kremer [edwin@cs.ruu.nl]

BIJLAGE H

NTG Software Distributie Service

Deze bijlage bevat een aantal distributie-services die door enkele NTG leden worden uitgevoerd.

1 TEX/LATEX distibutie VAX/VMS software

Date: Wed, 29 Nov 89 15:37:24 -0800
From: klerk@ijsapl.enet.dec.com
Subject: Distributie

Gerard,

Je kunt mijn naam en adres (p/a Digital Equipment bv, Europalaan 44, 3526 KS Utrecht) doorgeven als distributie centrum voor TeX op VMS. Evenwel heb ik geen laatste versie en doe ook geen pogingen steeds een nieuwe frutsel erbij te krijgen. De meest recente versie is degene van Johannes Braams (DECUS tape 1988) die inmiddels meen ik ook al weer enigszins is achterhaald.

Om een distributie te krijgen moet men mij een 2400' band sturen voor 6250 bpi (anders 2 banden) of een TK50 cassette. Hierop komen de RUNTIME.BCK en SOURCES.BCK tapes, alsmede de uitbreiding voor Nederlandse afbreekpatronen. Ik zal proberen ook Victor's Nederlandse artikel/report stijlen daarbij op te nemen nadat ik die van mijn Atari heb overgestuurd naar Digital's VAX.

TeX wordt door DEC steeds minder gebruikt (VAX DOCUMENT is ook voor mij de enig "zinvolle" versie ervan) en anders dan distribueren kan ik ook niet. Veel vragen m.b.t. installatie e.d. kan ik niet beantwoorden.

Theo de Klerk

2 TEX/LATEX distibutie UNIX software

Date: Tue, 31 Oct 89 18:19:56 +0000
From: Peter van Campen <petervc@SCI.KUN.NL>
Subject: Unix TeX tape distributie
Sender: TEX-NL <TEX-NL@HEARN.BITNET>

Vanaf nu distribueer ik Unix TeX tapes in Nederland. Het zijn copieen van een tape van Karl Kleine uit Duitsland, die ze weer uit van Pierre MacKay uit de USA krijgt. Het zijn directe copieen zonder toevoegingen of weglatingen.

Ik hoop steeds als Karl Kleine een nieuwe versie krijgt, zo snel mogelijk die nieuwe versie in Nederland te kunnen verspreiden.

Het gaat nu om TeX 2.99, LaTeX 2.09, METAFONT 1.7 op een tape in Unix tar-formaat als 1 file.

De regels zijn redelijk simpel:

1. Stuur mij een goede, ongemerkte tape (de volgorde is mijn voorkeur):
   - 1/2", 600 of 2400 ft reel (1600 bpi dan wel 6250 bpi)
   - 1/4" cartridge (QIC-11 of QIC-24)
   - tk50/tk70 cartridge

   met een sticker waarop het retour-adres staat, liefst in een herbruikbare verpakking.
   Hoe lager op de voorkeur-lijst, des te langer kan het versturen duren.
2. Je krijgt dan binnen ongeveer 2 weken een (in principe andere) tape terug waarop het staat. Als het in het begin erg druk wordt, kan het wat langer duren.

Peter van Campen
petervc@sci.kun.nl
Computer en Communicatie Zaken
B-Faculteiten
Katholieke Universiteit Nijmegen
Tournooiveld 1
6525 ED Nijmegen

3 TEX/LATEX distributie MS/PC-DOS software

Een Public Domain MS/PC-DOS demonstratieversie van TEX/LATEX is beschikbaar op NTG DOS-diskette NTG025+NTG026. Met deze diskettes heeft men een volledig functionele versie van TeX 2.98 in bezit. Gezien de omvang van een werkend TeX systeem is er bij het maken van deze demonstratie versie verondersteld dat men in het bezit is van een PC met harddisk (of een hiermee vergelijkbaar verwisselbaar schijfsysteem).

Vanwege de aanwezige previewer DVIEW wordt ook verondersteld dat het beeldscherm voldoet aan de CGA, EGA, VGA standaard. Voor een PC met een Hercules videokaart is het derhalve (helaas) nodig om gebruik te maken van een zogenaamde CGA-emulator, die onder verschillende namen rondzwerven (met de naam MG, SIMCGA of iets dergelijks). Deze files zijn op de tweede floppie (TEXDEMO_2) opgenomen. Voor gebruik met een Hercules videokaart zie de speciale opmerking bij gebruik.

Met vragen/opmerkingen over PC-versies van TeX, zo-

als deze demonstratie versie, kunt u zich wenden tot:

Jos Winnink,
Katwijkstraat 67-a,
2586 VL Den Haag
tel: 070-3514151 (overdag)
070-3554811 ('s-avonds)
internet: winnink@ecn.nl
earn: esu0507@hpeenr51

Electronic mail geniet de voorkeur boven brieven en telefoon.

4 TeXHaX herdistributie

Date: Mon, 20 Nov 89 16:35:57 -0100
From: Piet van Oostrum <piet@CS.RUU.NL>
Subject: texhax archief
Sender: TEX-NL <TEX-NL@HEARN.BITNET>

Wij hebben texhaxen vanaf dec 1986 gearchiveerd. Op experimentele basis zijn deze op te vragen via een mailserver en via FTP.

Mail-server: stuur een boodschap naar 'mail-server@cs.ruu.nl' met in de body bijv. het volgende:

send texhax/1989/090
end

Altijd 3 cijfers voor het nummer gebruiken.
De eerste keer kan het nuttig zijn om ook het commando 'help' op te nemen. Geloof evenwel nog niet alles wat er teruggestuurd wordt. De 'end' is alleen nodig als er troep achter kan komen.

FTP (anonymous): praxis.cs.ruu.nl (131.211.80.6), directory TEX/texhax

Problemen s.v.p. even melden per e-mail.

Het is de bedoeling dat er meer spullen komen te staan (style files, e.d), verzoeknummers worden in overweging genomen. Disk ruimte is voorlopig geen probleem.

Piet van Oostrum,
Dept of Computer Science,
University of Utrecht
Padualaan 14,
P.O. Box 80.089,
3508 TB Utrecht.
Telephone: +31-30-531806
Telefax: +31-30-513791
Uucp: uunet!mcsun!hp4nl!ruuinf!piet
Internet: piet@cs.ruu.nl

BIJLAGE I

NTG DOS-diskette Distributie Service

Distributie medium	:	5.25 inch, 360 kB floppy voor IBM-PC (DOS)
Kosten	:	f. 5,- per diskette
Verspreiding	:	op NTG bijeenkomsten (bij 3 of meer ook per post)
Contactpersoon	:	Gerard van Nes
Versie	:	october 1990

NGT001	312.320 bytes	:	TEXNL	TEX-NL communicatie	dec 88 t/m apr 89
			SOURCE	BRIDGE.TEX	Programma C.G. van der Laan (03 feb 89)
				SJABLOON.TEX	Idem (08 feb 89)
				CHESS.SHA	Schaakfonts P. Tutelaers (03 apr 89)
			NTG	VERSL88.01	NTG verslag 23 juni 1988
				VERSL88.02	NTG verslag 24 november 1988
				VRAGEN.TEX	Vragen lijstje NTG 2e NTG bijeenkomst
				QUEST.TEX	Questionnaire NTG
NTG002	343.040 bytes	:	TeXMaG		2.1 2.4 t/m 2.6 3.1 t/m 3.2
NTG003	338.944 bytes	:	TeXHaX		100.88 t/m 110.88
NTG004	338.944 bytes	:	TeXHaX		111.88 + 001.89 t/m 011.89
NTG005	349.184 bytes	:	TeXHaX		012.89 t/m 024.89
NTG006C	350.208 bytes	:	HYPPAT	VANROOSE.NOT	Informatie + Nederlandse Hyphenation file
				VANROOSE.HYP	van Peter Vanroose (2760 bytes)
				CELEX.NOT	Informatie + Noord-Vlaams Hyphenation
				CELEX.HYP	pattern CELEX/RUU (59758 bytes)
			STY	A4.STY	A4 style file N. Poppelier en J. Braams (12.03.90) versie 0.3
				A4.TEX	Bijbehorende documentatie; zie ook diskette NTG014 voor A4.DOC
				DUTCH.STY	Document style file (J. Braams; versie 1.09; 27.05.89)
				SOBER.STY	Document style file (N. Poppelier; versie 24.06.89)
				ARTIKEL1.STY	Zie TEX-NL filelist (1.14; versie 23.04.90)
				ARTIKEL2.STY	Zie TEX-NL filelist (1.12; versie 23.04.90)
				ARTIKEL3.STY	Zie TEX-NL filelist (1.15; versie 23.04.90)
				RAPPORT1.STY	Zie TEX-nl filelist (1.11; versie 23.04.90)
				RAPPORT3.STY	Zie TEX-NL filelist (versie 08.02.90)
				NTG10.STY	Zie TEX-NL filelist (versie 03.08.89)
				NTG11.STY	Zie TEX-NL filelist (versie 03.08.89)
				NTG12.STY	Zie TEX-NL filelist (versie 03.08.89)
				VOORWERK.STY	Zie TEX-NL filelist (0.5; versie 15.04.90)
			TEX	NTGSTYLE.TEX	Zie TEX-NL filelist (versie 04.09.89)
			LAYOUT	LAYOUT.TEX	Artikel "Layout-Anderungen mit LaTeX" van Hubert Partl
				LAYOUT2.TEX	bijbehorende file
				GERMAN.STY	idem
				REFMAN.STY	idem
NTG007	359.424 bytes	:	TeXHaX		025.89 t/m 037.89
NTG008	359.424 bytes	:	TeXHaX		038.89 t/m 051.89
NTG009	344.064 bytes	:	TeXHaX		052.89 t/m 067.89
NTG010	358.400 bytes	:	TeXHaX		068.89 t/m 084.89

NTG011	342.016 bytes	:	TeXHaX		085.89 t/m 100.89
NTG012B	355.328 bytes	:	TeXMaG		2.2+3.3+3.4
			TeXHaX		101.89 t/m 102.89
			TeXHaX	TEXHAX.89	Inhoudsopgave TeXHaX 1989 (89.001-89.102)
NTG013	360.448 bytes	:	TEXNL	TEX-NL communicatie	mei 89 t/m nov 89
NTG014B	354.304 bytes	:		INTRO.TEX	A gentle introduction to TeX; a manual for self-study (M. Doob; jan 1990)
				BRIEF.STY	Zie TeX-NL filelist (versie 07.02.90)
				BRIEF.TEX	Zie TeX-NL filelist (versie 24.10.89)
				BRIEFDOC.TEX	Zie TeX-NL filelist (versie 20.11.89)
				ASCII.TEX	Zie TeX-NL filelist (versie 20.10.89)
				ARTDOC.TEX	Zie TeX-NL filelist (versie 04.09.89)
				RAPDOC.TEX	Zie TeX-NL filelist (versie 04.09.89)
				BOEK.STY	Boek style versie (06.02.90)
				A4.DOC	Documentatie A4.STY; zie diskette NTG006
NTG015	352.256 bytes	:	TeXHaX		103.90 t/m 116.90
					001.90 t/m 003.90
NTG016	357.376 bytes	:	TeXHaX		004.90 t/m 020.90
NTG017	361.472 bytes	:	TeXHaX		021.90 t/m 037.90
NTG018	332.800 bytes	:	TEXNL	TEX-NL communicatie	nov89 t/m feb90
NTG019	188.416 bytes	:	NTG	NOTUUL3A/B/C	3^e NTG vergadering (11 mei 1989)
NTG020	271.360 bytes	:	NTG	NOTUUL4/A/B/C	4^e NTG vergadering (23 november 1989)
NTG021	358.400 bytes	:	STY	TUGBOAT.COM	Common commandos voor Tugboat styles
				TUGBOAT.STY	Plain TeX style voor Tugboat artikelen
				LTUGBOAT.STY	LaTeX style voor Tugboat artikelen
				SUPERTAB.STY	Tabular style van Jurriens/Kruljac
				SUPERTAB.TEX	Bijbehorende documentatie
				MULTICOL.STY	Multi column style van Frank Mittelbach
				WP2LATEX.xxx	WordPerfect to LaTeX translator
				CHANGBR.STY	Change bar style van Michael Fine
				CHBAR.STY	Change bar style van Dave Johnsen
NTG022	360.448 bytes	:	TEXNL	TEX-NL communicatie	mrt89 t/m mei90
NTG023	354.304 bytes	:	TeXHaX		038.90 t/m 054.90
NTG024		:	TeXHaX		055.90 t/m
NTG025	360.448 bytes	:	(La)TeX		Demo-versie (La)TeX + previewer (1/2)
NTG026	351.232 bytes	:	(La)TeX		Demo-versie (La)TeX + previewer (2/2)
NTG027	354.304 bytes	:	TEXNL	TEX-NL communicatie	jun90 t/m aug90
NTG028		:	TeXMaG		4.1 t/m 4.5

Enkele diskettes zijn updates (A, B,. . . nummering) van eerdere uitgaven. Degenen die in het bezit zijn van een oude versie kunnen deze zonder kosten omruilen.
De kosten van NTG001, NTG019 en/of NTG020 (Verslagen/bijlagen NTG vergaderingen) bedragen f. 3.00 per diskette.
Distributie van de diskettes is alleen voor NTG leden.

BIJLAGE J

Werkgroep 7: PC-zaken

TEX voor MS/PC-DOS PC's

Jos Winnink

September 21, 1990

Abstract

Searching for TEX implementations that were available for MS/PC-Dos PC's several *public domain* versions were discovered. These versions range from a system that only has TEX to a system that competes succesfully with commercially available implementations of TEX. Some indication of the speed of the TEX implementations is given.

Inleiding

Bij mijn poging een TEX implementatie voor de MS/PC-Dos PC's te krijgen bleken er meerdere *Public Domain* versies te bestaan.[1] De beschikbare versies liepen uiteen van één, die weinig meer biedt dan een (goede) implementatie van TEX tot één, die kan wedijveren met commerciëel verkrijgbare versies van TEX. Er werden ook een tweetal losse previewer's gevonden.

In dit artikel worden de gevonden implementaties beschreven. Tevens was dit *onderzoek* een gelegenheid om een vergelijking te maken tussen de snelheid van de verschillende TEX implementaties.

De volgende versies van TEX implementaties en *previewers* worden beschreven:

- DOSTEX
- DVIEW
- DVIVGA
- emTEX, 2 versies
- PUBTEX
- SbTEX, 2 versies
- TURBOTEX, een commerciëel verkrijgbare versie

DOSTEX

DOSTEX is een versie, die bestaat uit:

- $\mathcal{AMS}$-TEX
- LATEX
- TEX 2.93
- previewer voor Hercules beeldschermen (DVI2HERC)
- driver voor Epson FX-80 matrix printers (DVIEPS)
- een beperkt aantal fonts, zowel PK als TFM files

Deze versie is niet erg snel. Het systeem is zeker niet foutvrij, zowel DVI2HERC als DVIEPS vertonen een aantal problemen, waardoor er bijvoorbeeld bij het afdrukken van teksten, die wiskundige formules bevatten, soms iets fout gaat. Ook lijkt er een probleem te zitten in de implementatie van TEX, die zich manifesteert als er mathematische formules in de tekst aanwezig zijn. (TEX blijft dan soms hangen)

Beschikbaarheid: SIMTEL20 archieven

DVIEW

Previewer gemaakt op het MIT door Steve Ward en Ricardo Jenez. Heeft als bijzonderheid dat er slechts gewerkt wordt met fontfiles op de grootte 480 (96 DPI). Is geschikt voor CGA/EGA/VGA/Toshiba T3100/Olivetti M24 schermen.

Beschikbaarheid: SIMTEL20 archieven

DVIVGA

Dit is een previewer speciaal ontwikkeld voor EGA/VGA schermen. Het voordeel is dat ook de source code beschikbaar is. Het systeem heeft veel fonts nodig in tegenstelling tot DVIEW. DVIVGA is niet overdreven snel.

Beschikbaarheid: SIMTEL20 archieven

emTEX

Op dit moment is emTEX het paradepaardje onder de *Public Domain* versies van TEX voor MS/PC-Dos PC's. Het kan naar mijn mening wedijveren met commerciële versies.

De meeste recente versie van emTEX bevat de volgende componenten:

[1] Een demo-versie van TEX, LATEX en DVIEW is samengesteld; zie ook diskette NTG25 en NTG26 (bijlage I).

1. BIBTEX
2. Drivers voor matrixprinters (9 en 24 pins), HP Laserjet, HP Deskjet en screenpreviewer (CGA/EGA/VGA/HERCULES ....), POSTSCRIPT nog niet beschikbaar maar er wordt aan gewerkt
3. LATEX
4. MAKEINDX
5. METAFONT 2.0
6. PICTEX
7. PKedit
8. TEX 3.0
9. TEXCAD
10. WEB

Enkele uitbreidingen die emTEX duidelijk onderscheiden van andere implementaties:

- TEXCAD, een programma waarmee interactief figuren kunnen worden gemaakt en waarvan het resultaat een file is met instrukties voor de LATEX-picture environment.
- PKedit, een programma waarmee PK files kunnen worden *geëdit*.
- het gebruik van zogenaamde *fontlibraries* waarbij PK files in een *library file* kunnen worden opgenomen. Het grote voordeel hiervan is dat fragmentatie van de harddisk voor een belangrijk deel wordt voorkomen. De fragmentatie wordt veroorzaakt door het feit dat een file in werkelijkheid op een schijf een geheel aantal allocatieeenheden inneemt. Op een PC is de minimale allocatieeenheid op een vaste schijf 2 Kbytes. Als we uitgaan van bijvoorbeeld 100 PK file's per vergrotingsmaatstaf en we hebben de vergrotingen `magstep 0` tot en met `magstep 5` beschikbaar dan praten we over zo'n 600 file's. Gemiddeld geeft elke file een verlies van $\frac{1}{2}$ allocatie-eenheid, zodat het gaat om een ongebruikte diskruimte van ongeveer 600 KByte. Hierbij is geen rekening gehouden met ruimte, die de directories in beslag nemen.

emTEX bevat tevens diverse specifieke files voor het duitse taalgebied, zoals afbreekpatronen e.d.

Beschikbaarheid:

via FTP:

rusmv1.rus.uni-stuttgart.de	[129.69.1.12]	(in: soft/tex/emtex)
terminator.cc.umich.edu	[35.1.33.8]	(in: msdos/text-mgmt/TeX/emtex)
eba.eb.ele.tue.nl	[131.155.2.25]	(in: pub/tex/emtex.new)
sol.cs.ruu.nl	[131.211.80.5]	(in: pub/tex/emtex)

grote hoeveelheden bij deze laatste alleen tussen 18.00 en 8.00

floppy disks:

Pieter Bison	`BISON@KUNPV1.PSYCH.KUN.NL`
ondergetekende	`WINNINK@ECN.NL`

GUTenberg versie

De GUTenberg diskettes bevatten:

1. BIBTEX
2. TFM-files voor de zogenaamde D-fonts, in afwachting van oplossing van het fontprobleem (TEX3.0 is immers 8-bit geworden) worden bijgevoegd. Deze D-fonts worden via het *virtual font* mechanisme op C-fonts afgebeeld.
3. Drivers voor matrixprinters (9 en 24 pins), HP Laserjet, HP Deskjet en POSTSCRIPT (DVIALW). Screenpreviewers voor CGA, EGA, VGA, HERCULES (CDVI).
4. JOVE (Jonathans Own Version of Emacs), zoals de naam zegt een implementatie van de EMACS editor voor de PC.
5. LATEX
6. TE, een eenvoudige editor vergelijkbaar met de editor van TURBO PASCAL.
7. TEX 3.0, gebruikt wordt SbTEX 30.

Hoewel dit een aardige distributie is moeten in elk geval van elders de PK-files worden verkregen. Omdat METAFONT niet is bijgevoegd kunnen geen eigen fonts worden gegenereerd. Voor opmerkingen over SbTEX zie aldaar.

Beschikbaarheid: groep van franstalige TEXgebruikers (GUTenberg)

PubliCTEX

Implementatie van Klaus Thull (zie o.a.: TUGboat, Vol. 10, #1, p.15-22). Met PubliCTEX had ik problemen met installeren. Vanwege het feit dat het een implementatie is van TEX 2.93 alsmede de verhalen in TUGboat dat het niet zo'n snelle versie zou zijn heb ik pogingen om deze versie aan de loop te krijgen gestaakt.

Overigens is het systeem redelijk geschikt als bron van allerlei TEX zaken, zo bevat het ondermeer de volgende componenten:

Implementatie:		**TEX versie**	**Relatieve snelheid**
MS/PC-DOS:			
	DOSTEX	2.93	0.13 – 0.29
	TurboTEX(1.1a)	2.92	0.13 – 0.27
	PubliCTEX	2.93	?
	SbTEX26	2.98	0.84
	SbTEX30	3.0	0.87
	emTEX	2.99[2g]	1.00
	emTEX (80286 versie)	2.99[2g]	1.04
	emTEX	3.0[3a]	1.14
	emTEX (80286 versie)	3.0[3a]	1.16
Overige:			
	VAX 750 (DEC distr. tape)	2.93	0.75
	SUN 3/60 (P. v. Campen tape)	2.99	3.92
	SUN/SPARC (P. v. Campen tape)	2.99	13.27

Table 1: Snelheidsvergelijking diverse TEX implementaties

1. *CHEMSTRUCT* van Dr. Michael Ramek
2. DVISELEC, een programma waarmee delen uit DVI-files kunnen worden geselecteerd.
3. EPIC, de Extended LATEX picture environment.
4. LATEX
5. METAFONT source's voor o.a. het lettertypes OCRB en PUNK
6. PICTEX
7. TGRIND, een programma dat listings van programma's in diverse programmeertlen *pretty* kan printen, door te functioneren als *preprocessor* voor TEX.
8. TEX 2.93
9. WEB

Beschikbaarheid: Heidelberg server

SbTEX

SbTEX is ontwikkeld door Prof. Wayne Sullivan in Dublin. Het is een relatief snelle versie maar is beperkt tot enkel TEX. Voor een werkend systeem is veel meer nodig, GUTenberg gebruikt SbTEX 30 als basis voor het door hen gedistribueerde TEX systeem.

Beschikbaarheid: SIMTEL20 archieven

TurboTEX

Over deze implementatie staat mij weinig informatie ter beschikking. Eigenlijk is TurboTEX enkel en alleen opgenomen vanwege de vergelijking in snelheid tussen de verschillende implementaties. Opmerkelijk is overigens dat de geteste versie van TurboTEX erg gevoelig bleek te zijn voor de hoeveelheid beschikbaar geheugen. Wanneer in plaats van 535 Kbyte *slechts* 490 Kbyte vrij geheugen ter beschikking stond dan nam de verwerkingstijd toe met een factor twee.

Beschikbaarheid: commercieel pakket

Snelheidsvergelijking

Om enigszins een gevoel te krijgen voor de snelheid van de verschillende TEXimplementaties is besloten om hetzelfde document telkens door de verschillende implementaties te laten zetten. De resultaten zijn vermeld in tabel 1. Hierbij moet onmiddelijk worden opgemerkt dat het hier niet gaat om een uitgebreide en uitgebalanceerde *benchmark*, maar slechts om een test met de bedoeling om een indicatie van de verwerkingssnelheid van de verschillende implementaties te geven.

Als document is gekozen voor de documentatie van het macro pakket *CHEMSTRUCT* van Dr. Michael Ramek. De overwegingen om dit document te kiezen waren ondermeer:

- de omvang van in totaal 18 pagina's, zodat effecten van het laden van macro's, wegvallen ten opzichte van de benodigde CPU-tijd.
- het feit dat er ruimschoots gebruik gemaakt wordt van macro's om structuurformules van chemische verbindingen te zetten.
- het feit dat *CHEMSTRUCT* bedoeld is om te werken in combinatie met *PLAIN*, zodat eventuele geheugenproblemen, die soms optreden wanneer LATEX wordt gebruikt in combinatie met een ander macropakket, worden voorkomen.

Een enkele opmerking is op zijn plaats. De snelheid van emTEX 2.99[2g], die op 1.0 wordt gesteld, komt overeen met een verwerkingstijd van ongeveer 540 seconden op een TULIP COMPACT II, een XT achtige PC met een 10 Mhz V20.

De vergelijkingen met de VAX 750, de SUN 3/60 en de SUN/SPARC zijn toegevoegd om enigszins een indruk te geven van de prestaties van de PC implementaties ten opzichte van andere machines. Van een echte benchmark (wat dat dan ook moge zijn) is *geen* sprake.

Conclusie

Voor mijzelf is de conclusie inmiddels duidelijk. Nadat ik eerst geprobeerd heb om uit diverse distributies een goed werkende versie van TEX samen te stellen incusief drivers voor printers en beeldscherm alsmede programmatuur als BIBTEX, MKINDEX e.d., bleek het moeilijk te zijn om een compleet werkend systeem te verkrijgen. Dit werk is te vergelijken met het samenstellen van de GUTenberg diskettes.

emTEX daarentegen blijkt een complete implementatie te zijn inclusief METAFONT met een zeer goede coherentie tussen de verschillende delen van het systeem.

Alles overziende lijkt op dit moment emTEX de grote winnaar. Het pakket is compleet in die zin dat er weinig ontbreekt en op het gebied van snelheid is het ongeslagen. Of er commerciële versies beschikbaar zijn die beter voldoen is niet bekend, maar men moet van zeer goede huize komen om beter uit de bus te komen dan de *gratis* versie van emTEX.

emTEX aangevuld met enkele ontbrekende delen zoals $\mathcal{AMS}$-TEX, PICTEX en de NTG style file's kan dan ook dienen als de NTG TEX3.0 distributieversie voor MS/PC-Dos PC's.

BIJLAGE K

Werkgroep 8: SGML-TEX Seminar

Terugblik op organisatie

Kees van der Laan

De voorbereiding kan gekenschetst worden door delegatie en het herhaald terugkoppelen met betrokkenen. Het begon met het aftasten van het idee een SGML-TEX-seminar te laten plaatsvinden. Dan de samenstelling van de commissie, de opzet, selectie van plaats van handeling, taakverdeling, tijdsplanning, etc. Met tenslotte de diverse terugkoppelingen bij veranderingen, c.q. verfijningen, en niet in het minst het uitvoeren van het vele werk.

De commissie bestond uit twee NTG-ers (Kees van der Laan, Theo Jurriens) en twee SGML-ers (Jan Maasdam, Jan Bleeker) als basis. Bij de uitwerking van de opzet zou blijken dat nog enige taken gedelegeerd moesten worden.
De opzet was een 1-daagse seminar te houden over SGML en TEX etc, met cursussen in de week ervoor en erna. Deze basiszaken werden in het najaar van '89 met de besturen en achterbannen besproken. Ook werden de benodigde zalen gereserveerd en principe-afspraken met de kantinedienst gemaakt in verband met lunches en koffie/thee pauzes. In december '89 kwam een opzet klaar m.b.t. cursussen, de locatie, de indeling van de conferentiedag. Ook werden twee raambegrotingen opgesteld: één voor het seminar en één (sjabloon) voor de cursussen. Dit geheel werd besproken met de besturen: NTG (december) en SGML Holland (februari). Elsevier en Samsom hebben toen aangeboden de deelnemers aan het seminar te onthalen op een receptie.
Op grond van de schatting van het aantal deelnemers werd de prijs vastgesteld voor deelname aan het seminar en voor deelname aan de diverse cursussen. Er werd onderscheid gemaakt in de prijs voor leden (ook van zustervereniging en) en niet-leden. Voor cursussen werd tevens gelet op de internationaal gehanteerde prijsstellingen.) Met de voorgestelde taakverdeling en tijdsplanning werd, na her en der een kleine aanpassing, ingestemd.

Taakverdeling (in hoofdlijnen):

Commissie	Kees van der Laan, Theo Jurriens, Jan Maasdam en Jan Bleeker.
Financiele zaken	Koen Mulder.
Programmaboekje	Gerard van Nes.
PR-zaken	Jan Maasdam, Jan Bleeker.
Demos (Leveranciers)	Theo Jurriens.
Locale logistiek	Kees van der Laan.

Rond dezelfde tijd is ook het RC-RUG geïnformeerd over de mogelijke cursussen, en is overeengekomen nadere afspraken later te maken over:

- welke cursussen precies,
- het aantal deelnemers,
- de gewenste apparatuur,
- de installatie van de te gebruiken programmatuur, dit in nader en direct overleg met de docent.

(TEX etc. programmatuur was reeds operationeel op de VAX en een campuslicentie voor μ-TEX was in het bezit van RC-RUG. Sobemap zou benaderd worden voor het verkrijgen van het gebruiksrecht van hun parser door SGML Holland/MID).

Rond de jaarwisseling werd een 'Call for papers' uitgezonden naar de achterbannen en op de diverse electronische netten, TUGboat, e.d. Veel moeite is besteed aan het benaderen van de diverse (buitenlandse) sprekers.

De organisatie werd bemoeilijkt door ziekte van ondergetekende en doordat Jan Bleeker van werkkring veranderde. Jan Maasdam heeft contact met het RC-RUG ondernomen en zich van de doorgang en steun vergewist. Dieke van Wijnen nam de taken van Jan Bleeker over. Kees van der Laan zou thuis doorgaan met zijn organisatietaken.

Later toen het programma voor het seminar vaststond, is dit, inclusief vermelding van de te houden cursussen, verspreid aan de diverse achterbannen (ook hebben de diverse TEX-gebruikersgroepen in Frankrijk, Engeland en Duitsland hun leden geïnformeerd), verspreid over de diverse e-mail netwerken en is er een persbericht uitgegaan. Ook zijn er diverse tijdschriften benaderd voor plaatsing van een annonce c.q. redactioneel stukje. SGML Bulletin en TUGboat hebben de annonces geplaatst. In een nog later stadium heeft MID in Duitsland voor verspreiding van de diverse aankondigingen gezorgd.

Rond juni zijn er beslissingen genomen door de respectievelijke besturen, over het al dan niet doorgaan van enkele cursussen waarvoor weinig aanmeldingen waren binnengekomen. Wegens het geringe aantal aanmeldingen voor LATEX beginners, LATEX stijlen, en Metafont, zijn deze cursussen geannuleerd. TEX advanced zou (eventueel) gesubsidieerd worden door de NTG; ook werd het honorarium in overleg met de docent, bijgesteld. Om meer deelnemers te trekken, m.n. uit Duitsland, is de conferentietaal rond die tijd gewijzigd in Engels.
Al hoewel in principe hotelreserveringen door deelne-

mers, docenten en sprekers zelf geregeld zouden worden, zijn er enige bemiddelingen geweest, m.n. voor de Amerikaanse sprekers.
Op 28 augustus kon er niet van de standaard voorzieningen gebruik gemaakt worden en moesten er aparte regelingen getroffen worden.

Na het seminar is er een informeel diner gehouden voor de sprekers en de betrokkenen bij de organisatie ter afsluiting.

Enige cijfers

Aantal deelnemers seminar (incl. sprekers):
90 (5 Amerikanen, 2 Engelsen, 8 Duitsers, 1 Zwitser)
Deelnemers cursussen:

14	SGML orientatie	(28 aug)
9	SGML passief	(29–30 aug)
5	SGML actief	(3–4 sept)
16	TEX intermediate	(28-30 aug)
5	TEX advanced	(3–5 sept)

Opmerkingen

De toegang tot de VAX op 28 en 29 augustus ondervond enige problemen omdat de usernummers/passwords niet geaccepteerd werden. (De dag ervoor was onder *docenten*nummer/password de werking geverifieerd.) Ook was het niet altijd mogelijk uitvoer naar de laser printer te 'routen.' Eén van de TEX docenten heeft schriftelijk zijn ontevredenheid betuigd over de logistiek rondom zijn cursus.

Leveranciers demos

Arbortext, MID (Duitsland), TEXnology Inc., en de Universiteitsdrukkerij lieten resultaten zien verkregen met Varityper 600, en helaas was BYTE er niet vanwege problemen met de ATARI TT. Verder verzorgde Wristers/Scholten de boekenstand. Speciaal ter gelegenheid van deze bijeenkomst hadden NTG-ers een PD-MS-DOS tegen floppykosten beschikbaar gesteld.

Tot Slot

De financiële afwerking is nog niet rond en zal door en in overleg met de penningmeesters afgehandeld worden.

De (La)TEX bronteksten van annonces, begrotingen, planning, programmaboekje e.d. zijn op floppy beschikbaar voor het volgend organiserend committee.

BIJLAGE L

Working Group 13: 'Neerlandica'

Vergadering 18 april 1990

Nijmegen

Inleiding

Het begint traditie te worden dat werkgroep 13 enkele weken voor de halfjaarlijkse vergaderingen van de NTG bij elkaar komt. Het gebruikelijke gezelschap, Johannes Braams (PTT Research Neher Lab), Nico Poppelier (Elsevier Science Publishers) en Victor Eijkhout (Katholieke Universiteit Nijmegen, coördinator en notulist), was deze keer (eenmalig?) uitgebreid met David van Leeuwen (Rijksuniversiteit Leiden).

Werkgroep 13 mag reeds na een jaar terugzien op een roemrijk verleden. De vervangingsstijlen voor LaTeX, en de compleet nieuwe briefstijl zijn reeds een feit (hierover is gerapporteerd in TUGboat [1]); een eerste versie van de taaloptie `dutch` was al beschikbaar, en het werk aan de stijloptie `a4` is voltooid (rapportage hierover volgt in TUGboat [2]).

Het voornaamste programmapunt was deze keer de nieuwe versie van `dutch`, die deel uitmaakt van een groter project, `babel` genaamd; het werk van Johannes.

Taalspecifiek: Dutch

De taaloptie `dutch` bevatte reeds parameterwaarden voor taalafhankelijke delen van de standaardstijlen van LaTeX – dit omvat ook de `\date`. Sinds de vorige keer is daar een stuk bijgekomen dat de trema actief maakt; zie de notulen van de vorige vergadering.

Nico had hier grotendeels de macro's uit de stijloptie `german` – zie [3] – voor gebruikt. David had wat kritiek hierop, en al discussierend kwamen nog wat bezwaren aan het licht.

- De gebruikte benaming 'umlaut' is niet optimaal: 'trema' is beter.
- Nico tekent aan dat TUG aanraadt in publicaties over TeX de naamgeving uit de boeken over TeX en LaTeX te hanteren. In het TeX-boek wordt de Nederlandse trema aangeduid met 'dieresis or umlaut accent'. De term 'unlaut' is in zijn ogen dus niet misplaatst.
- De trema moet niet actief zijn in wiskunde-modus.
- Het is niet nodig uitgebreid te testen op welk teken de trema komt: er zijn uitsluitend enige tests nodig voor de speciale gevallen. In principe is een trema door
  `\discretionary{#1}{-}{\"#1}`
  te implementeren.

Nico of Johannes zal deze bezwaren verwerken.

Voor de Nederlandse 'aanhalingstekens openen' worden nu de macro's uit `german` gebruikt. Eigenlijk zou dit een aparte ligatuur moeten zijn.

Verder kan `dutch` nu vanuit Plain TeX gebruikt worden.

Taalonafhankelijk: Babel

In overleg met Hubert Partl is Johannes bezig geweest met een poging standaardisatie te bereiken van nationale aanpassingen aan de LaTeX stijlen.

Resultaat hiervan is een stijloptie `babel` die het gemeenschappelijk werk doet van alle nationale aanpassingen. Deze optie wordt automatisch geladen door de verschillende taalopties, en hierin worden de herdefinities van talige termen gedaan. Voorts bevat deze optie het mechanisme om nieuwe talen te kunnen definiëren.

Afzonderlijke taalopties hoeven nu enkel de parameterwaarden te zetten, en kunnen zelf zaken als de actieve trema – zowel in `german` als in `dutch`, maar onderling op enkele punten verschillend – regelen.

Het ziet er allemaal fraai uit en er was geen echte kritiek op. Kritiek geuit tijdens de vorige vergadering is allemaal verwerkt.

Overige punten

Over de optie `a4` is niet gesproken. Na het artikel van Johannes en Nico moet dat punt afgehandeld zijn. Ook de NTG stijlen zijn nauwelijks genoemd. Wel is de noodzaak uitgesproken in het algemeen goede gebruikersdocumentatie voor alle stijlen en opties te verzorgen.

Er komt een artikel door Johannes en Nico over het Babel-project in een nummer van TUGboat in 1990.

Er is, naar aanleiding van opmerkingen van David, gesproken over aangepaste fonts voor het Nederlands. David heeft enkele nieuwe ligaturen gemaakt, met name de 'fk' en de 'fij' ligatuur. Voor allebei geldt dat er vraagtekens te zetten zijn. Bij voorbeeld: de combinatie 'fk' bevat een lettergreepscheiding. Mag een ligatuur dan nog?

Verder zijn het voornamelijk practische problemen van standaardisatie die wijder gebruik van deze ligaturen ver-

hinderen. Een verder punt is dat David deze tekens in codes boven de 127 gestopt heeft, iets waar veel printeraansturingsprogramma's niet tegen kunnen. Dit zou opgelost kunnen worden door – wat David ook al gedaan heeft – enkele weinig gebruikte tekens uit de Computer Modern fonts als de 'æ' en 'œ' te vervangen door de 'fk' en 'fij' ligaturen. Met TEX 3.0 en de bijbehorende familie printeraansturingsprogramma's van Nelson Beebe moeten alle problemen hoe dan ook van de baan zijn.

Conclusie

Werkgroep 13 kan zichzelf nog niet opheffen, maar als de huidige projecten afgerond zijn zal er niet veel meer te doen zijn dan onderhoud plegen – althans, dat is wat de werkgroep voorziet. Victor is op deze toestand al een voorschot aan het nemen: hij heeft in het afgelopen half jaar enkel reparaties verricht aan de vervangingsstijlen.

Overigens worden de activiteiten van werkgroep 13 in dit tijdsgewricht wel interessanter (lees: lastiger) gemaakt door het feit dat zowel TEX als LATEX een revisie ondergaan. Misschien kan werkgroep 13 dus nog wel enige tijd vooruit.

References

[1] Johannes Braams, Victor Eijkhout en Nico Poppelier, *The development of national LATEX styles*, TUGboat 10 (1989) #3, 401–406.

[2] Johannes Braams en Nico Poppelier, *Babel, a multilingual style-option system for use with the standard document styles of LATEX*, te verschijnen in TUGboat vol. 11 (1990).

[3] Hubert Partl, *German TEX*, TUGboat 9 (1988) #1, 70–72.

BIJLAGE M

SGML-TEX conference
August 31, 1990
Groningen, The Netherlands

David Osborne

Buses and wierdness in Groningen

For their second full-day international meeting, the Nederlandstalige TEX Gebruikersgroep (NTG) organised, in conjunction with the Dutch SGML Users Group, a conference intended to focus interest on the use of TEX and SGML together. On August 31st, approximately 100 delegates from both 'camps' attended the day's events in Groningen, with what seemed a good balance between SGML-ers and TEXies.

TEX– and SGML streams

The only drawback was that most of the day was organised in two parallel streams of presentations and these tended to be a 'TEX stream' and an 'SGML stream'; consequently, delegates often chose to hear talks on the package with which they identified more closely – at least, this delegate did. So, one came away with a still hazy appreciation of the alternative approach to mark-up. I hope that this situation will improve in future and that staunch supporters of SGML and of TEX will each come to understand why their opposite numbers are so excited by *their* favourite method of tackling the mark-up problem. With that said, you may forgive me if the following account concentrates on TEX-related issues. Malcolm Clark, who was also at the conference has added some accounts (in square brackets) where he went to a different presentation.

Vendors

[There were some demonstrations from vendors, chiefly *The Publisher* from both TEXcel and MID Information, both running on Sun workstations. The local bookshop also had an impressive display of books for sale at the conference.]

Gerard Kempen

The day started with an introduction by Kees van der Laan, who, as President of NTG, had been instrumental in organising the meeting, with the assistance of members of the NTG and the SGML-Holland users group. In the opening talk, prior to the parting of the ways into the parallel sessions, Prof. Gerard Kempen, of the University of Nijmegen, discussed *Language technology and the future of text automation.* This was an account of some of the work in the Language Technology Project at Nijmegen, and focussed on methods for enhancing the transportability of text between systems including SGML, TEX and ODA/ODIF by separating content and form, logical structure and layout. The approach used employs object-oriented methods for parsing the syntax of sentences. Prof. Kempen described the use of 'Corrie', a module for checking and correcting Dutch spelling, which is based on the syntactic and morphological parsing of text. Other modules used in the project allow the inflection of known words and attempt to detect typos in unknown words, by analysing spelling and missing letters. Sentences are also scanned for idioms. In this way, parse trees are built which describe the hierarchical structure of sentences, storing not the words themselves but pointers to a standard lexicon. In the current implementation, it is possible to handle the rearrangement of subtrees graphically by means of a WIMP interface. It is hoped to develop a 'grammar spreadsheet' to allow the automatic propagation of word changes; for example, pluralising one word will result in appropriate changes in other parts of the sentence. The advantages of the approach taken in the project are claimed to be easier editing, enforcement of spelling standards, and applications including semi-automatic translation into other languages as well as speech, information retrieval, and hypertext.

Joop van Gent

As the first talk in the TEX stream, Joop van Gent, of the Institute of Language Technology and AI, Tilburg, spoke on *Two Faces of Text*, describing document retrieval and TEX. He described an approach to the structure of published documents which considered two types of structure: logical structure and typography. While automatic methods can be used for scanning the logical structure of a document, human readers tend to use typographic cues in the typefaces and layout of a document to locate information. It is difficult to imitate this search strategy in computer systems since it requires pattern recognition on the typeset page image. Only the typesetting software can 'know' the relationship between logical parts of a document and the typeset output. Work

is underway to link these by storing the positions of logical page-elements. At Tilburg, this has been done by adapting TEX to produce a logical structure tree and predicates in a format which can later be searched to locate parts of a document on the typeset page. The documents must be parsed while processing a search query, but the query language, EL/DR, can be interpreted or compiled by Prolog, allowing the use of natural language queries. Some knowledge of the document structure, as nested lists or tree structures with well-defined properties, is used to make basic operations trivial.

Malcolm Clark

The next speaker was Malcolm Clark, who addressed *Exchanging documents: a TEXnical perspective*, briefly reviewing document 'views' – typographic versus logical structure; content, which may not be orthogonal to the structure; and revisable versus non-revisable documents – and the requirements for document exchange, either within or between groups, perhaps over heterogeous networks. In the TEX community, the latter need is met to a greater or lesser extent by the use of electronic mail as a transmission method, and Malcolm discussed some of the problems which can arise. With the new version of TEX, providing support for 8-bit character sets, source files for documents must be somehow encoded into 7-bit data for transmission using *email*, and Malcolm advocated the provision of a 7-to-8-bit filter, written in WEB, as a standard tool in the TEX toolbox. This would enable the exchange of TEX source files, representing revisable documents, and DVI files, when compressed and encoded to ASCII, as non-revisable documents.

Since Malcolm had not used all of his allotted time, he was persuaded by Kees van der Laan briefly to discuss the use of graphics in TEX, to fill the remaining minutes before lunch.

Frank Mittelbach

Frank Mittelbach had been scheduled to conduct a 'work lunch' session on the new features in LATEX 3.0, but his talk was instead re-arranged into a third mini-stream at the start of the afternoon session. I chose to attend that, since it is more relevant to my work, thereby regretfully missing Amy Hendrickson's talk on TEX macro techniques, which I look forward to hearing at the Cork meeting. With Frank's talk over-running, I missed most of Victor Eijkhout's discussion of *The document style designer as separate entity*, but heard Johannes Braams' description of *The Dutch national LATEXeffort*. This included a discussion of replacements for the `article` and `report` document styles to produce documents with a more European appearance than the LATEX book. A new contribution from the Netherlands in this area is 'Babel', a multilingual document style for the standard LATEX styles. This replaces the language-specific headings, hyphenation patterns and explicit hyphenation points with support for more than one language in a document and allows easy switching between them. Usable with plain TEX or LATEX, each language can have a separate option file. When used with LATEX, the language can be selected on the `\documentstyle` command.

Sake Hogeveen

Sake Hogeveen, *Aspects of Scientific Publishing*, is an astronomer who also works with some publishing houses. He claimed to be presenting an introductory course in TEX in five minutes and a LATEX introduction in six! This seemed a bold claim which I could not resist. Basically he tried to argue through the markup route from genuine' copy editor mark up through to TEX typographic markup and then to structural markup. No great conceptual leaps there, but since it was addressed to an audience made up mainly of SGML people – that is, suits and ties – it had a useful function. One of the things to be learned fromn this meeting was the almost complete ignorance of the TEX world for SGML, and the equal ignorance of the SGML world for TEX and LATEX. Neither position is helped by our inability to speak to one another at the same level. Sad. Sake contended that good typography supports structure. Few could argue with that. He also seemed to be advocating that we should be prepared to get our fingers dirty by changing style files. This seems like a bad idea to me. LATEX style files are tortuous to 'amend': it often seems to me that this is a good thing. The average document producer is not a document designer, and his or her results can be appalling.]

Victor Eijkhout & Andries Lenstra

Victor Eijkhout & Andries Lenstra described a system, written in TEX which allowed a 'document style designer' to develop styles, without having to know TEX. They use a fairly free syntax to define constructs, placing the designer in a position between format writer and document producer. It was not at all clear to me whether this existed as a real product, or whether it was a set of stubs, full of good intentions. But even as a prototype it introduced many ideas which could be fruitful.]

Barbara Beeton

A fascinating trip down memory lane was then given by Barbara Beeton, who described ten years of *TUGboat production: TEX, LATEX and paste-up*, with a look at the pitfalls and rewards of editing TUG's journal, interspersed with sideways glances at fragments of TEX hagiography. A rich goldmine for devising a TEX trivia quiz!

Manfred Krüger

The streams were finally drawn together for a talk by Manfred Krüger of MID, who discussed *SGML and*

TEX: Two core modules of information processing. He highlighted the misunderstandings that SGML-ers and TEXies have of each other's worlds, claiming that the question, 'Which is better?', is not a useful comparision, since SGML 'does nothing', yet TEX is 'a solution for everything.' Publishing requires the use of SGML *and* TEX. A slightly heated discussion between speaker and audience resulted when it was suggested that the coupling of SGML marked-up documents with the new LATEX could be achieved by making LATEX macros read the SGML reference syntax, since this could allow an interface for the specification of processing instructions in a structured and standard way (cf. the results of the DAPHNE project).

Richard Southall

At the conclusion of this talk, the faint-hearted departed for the bar and the cocktail party. However, those with a thirst for knowledge who stayed were rewarded by a discussion of *Presentation rules and rules for composition in the formatting of complex text* by Richard Southall, now of Xerox EuroPARC. Richard began by demonstrating that both SGML and LATEX have missed aspects of the highest quality typesetting which can be achieved manually. While generalised markup can define 'content objects', that the presentation rules define the layout is only *sometimes* true. He then took examples from entries in *Math Reviews*, published by the AMS, having been recently involved in a complete redesign of the layout and typography. The markup in the MR entries is almost exhaustive, yet the entries before the redesign project show 'wierdnesses' in the typography since presentation rules are being applied to the whole entry, both in the body of the review proper and in the bibliographic details. Former colleagues from Reading University made a structural analysis of the whole journal, while Richard concentrated on the structure of individual entries. He found examples of poor spacing in authors' and journal names in the bibliographic references, leaving large spaces ('buses'. . . large enough to take a London omnibus) which produce barely perceptible interruptions when a reader scans the line. Since these entries frequently contain abbreviations and punctuation, he offered quotations from typographer's handbooks by Tchischold, DeVinne and others illustrating the presentation rules for punctuated text. These seemed to indicate that TEX's so-called 'french spacing' should be applied in such circumstances, since traditional spacing between sentences is that of normal word spacing. Lastly, the concept of 'compositional environments', in which a set of presentation rules apply, was discussed. In electronic formatting systems, these seem to have first been supported in *Scribe*, which provided delimited areas for such rules. In TEX, there are two clear compositional environments (modes): inline and display math. In other environments, implied or procedural rules are used, such as the spacing in quoted quotes, for example. The problems in the first TEX version of *Math Reviews* were mainly due to the application of the same presentation rules over all the content objects in each entry. Once appropriate compositional environments were introduced, it was interesting to see from the finished examples that the entries were more readable, yet occupied less space. Following a short question and answer session, and with our thirst for knowledge, if not beer, now slaked, we retired to the bar to join the end-of-conference festivities.

(Note added by editor)
Course along the meeting:

- SGML orientation (1 day)
- SGML passive (2 days)
- SGML active (2 days)
- TEX intermediate (3 days)
- TEX advanced (3 days)

- Arbortext, and Krüger MID
- TEXnology Inc.
- Bookstand Scholten/Wristers
- Groningen University Press showed products made by Varityper 600.

Program TEX & SGML Conference

9:00 Welcome, registration, coffee

10:00 Opening
10:15 •**A1**• Language technology and the future of text automation
Gerard Kempen (KUN)

11:00 Coffee

	Room ZG15	**Room ZG8**
11:30	•**T1**• Two faces of Text[1] *Joop van Gent (KUB)*	•**S1**• SGML at the European Patent Office *Terry Rowlay (EPO)*
12:15	•**T2**• Exchanging documents: a TEXnical perspective *Malcolm Clark (ICRF)*	•**S2**• Aspects of a marriage: SGML and TEX - an example from real life *Tibor Tscheke (Univ. Wurzburg)*

13:00 'Work'lunch Lunch
•**T7**• Towards LATEX 3.0[1]
Moderator: Frank Mittelbach & Rainer Schöpf

	Room ZG15	**Room ZG8**
14:00	•**T3**• Getting TEXnical: Insights into TEX Macro Writing Techniques[1] *Amy Hendrickson (TEXnology Inc.)*	•**S3**• Aspects of Scientific Publishing *Sake Hoogeveen (UvA)*
14:45	•**T4a**• The document style designer as separate entity[1] *Victor Eijkhout, Andries Lenstra (KUN)*	•**S4**• SGML Applications in the Building and Construction Industry *Hin Oey (TNO)*
15:15	•**T4b**• The Duth national LATEX effort[1] *Johannes Braams (PTT Research Neher Lab.)*	

15:45 Tea

	Room ZG15	**Room ZG8**
16:00	•**T5**• TUGboat production: TEX, LATEX and paste-up[1] *Barbara Beeton (TUG/AMS)*	•**S5**• SGML and TEX at Elsevier Science Publishers[1] *Jeroen Soutberg (ESP)*

16:45 •**A2**• SGML and TEX: Two core modules in information processing
Manfred Krüger (MID)

	Room ZG15
17:30	•**T6**• Presentation rules and rules of compo- sition in the formatting of complex text *Richard Southall (Rank Xerox)*

17:30 Cocktail party

All day: Vendor/Demonstration booths

[1] Conference material included in this report

BIJLAGE N

Two faces of text

TEX as a programming tool for advanced document retrieval systems

Joop van Gent

Institute for Language Technology and Artificial Intelligence (ITK)
Tilburg University
PoBox 90153
Tilburg, The Netherlands
gent@kub.nl

Introduction

It is a common misconception to think that a scientific book about earthquakes is a book about earthquakes. If this would be the case the book could be replaced by a database containing the *information about earthquakes* exposed in the book. All scientific books about earthquakes could be replaced by one such database and — still more improbable —all books in the world could be replaced by one huge knowledge base containing all ideas exposed in all books. This is not the case.

A scientific book about earthquakes is also a book about the author, the time of writing, the origins of the ideas and linked ideas that are stored in other books. It's a book about other books about earthquakes and a book about itself. The book presents a lot of different sorts of information, the organisation of which is taken care of by historically developed and commonly accepted methods of referencing (bibliographies, index tables, thesauri) and methods of division (logically splitting up information in parts like nesting paragraphs in sections, sections in chapters, etc.). People find their way in libraries and books by search strategies based on these finegrained organisation systems.

The basis of the main ideas presented in this article is that — *as long as hard copy documents exist parallelly to electronic ones* — document retrieval systems should not only make use of these advanced organisation methods but should also have user interfaces that allow search strategies that people are acquainted with. Document retrieval systems should pair the advantages of fast search performance of computers with 'traditional' bumming around in libraries and 'sniffing' in books.[1] This implies that users should have electronic documents available in a typeset format that looks like a corresponding hard copy as much as possible. It also implies that these systems allow users to query the database in *natural language.*

For the former statement there are several arguments. The most important one is probably the following. In contrast with searching in large databases and quickly overviewing pages, reading large pieces of text is in many ways more comfortable on paper than on a computer screen. Evidently, hard copies are also more easily carried in a handbag. It is therefore to be expected that hard copies (especially large ones) will keep playing a role together with electronic documents for quite a long time. To allow a user to comfortably switch from hard copy to screen and vice versa is therefore a prerequisite for a document retrieval system. This can be achieved by an 'isomorphism' of typography.[2] Another argument is that besides using referencing and division a user often finds his way by *recognition of patterns* without really reading text, by just running a thumb through the pages of a book. This 'method' is evidently supported by having pages in the same typesetting on the screen.

I don't have strong arguments for qualifying interrogation in natural language as an important feature of document retrieval systems. Nevertheless I think document retrieval is one of the 'discourse domains' where a natural language interface is pretty efficient in comparison with alternatives like menu structured interaction or command languages. This is so, because the objects by their nature have an extremely deep hierarchical structure. Therefore menu structuring makes search actions last unnecessary long and command languages — impopular as they are anyway — too complex.

In this article I want to show the important role Knuth's typesetting language TEX — as defined in [6] — can play in the kind of document retrieval systems introduced above (from now on I will call these systems *advan-*

[1] And, of course, with the advantages of new strategies, like those developed in Hypertext(-like) environments (see [3] for a short introduction & survey).

[2] In some sense the physical sectioning of documents in numbered pages can be seen as part of the typesetting. Switching between hard copy and screen is evidently aided by a correspondence in paging and pagenumbering.

ced document retrieval systems). But before doing this I will give a more detailed description of these systems.

Advanced document retrieval systems

Until recently document library systems have been limited by the fact that most documents are only available in (good old) hard copy format. For this reason it was only possible to *automatically* retrieve information about a document by looking at its *description* in a database. These descriptions were (and still are) stated in terms of attributes like author, date, editor, title, keywords etc., like the features one finds on the little paper cards in a library catalog. In other words: only the library *catalog* could be automatized. Most conventional database systems are designed to query 'things' that are not really in the database. For example a database of second hand boats does not really contain second hand boats, it only contains *descriptions* of them.

Since more and more documents become available in (some) electronic format, it becomes interesting to develop library systems with databases containing electronic documents. What makes a database of electronic documents so special is that it does really *contain* electronic documents! Retrieval systems therefore can have the property of being able to *show* the objects in the database and we probably want to use this ability: we want to be able to treat a lot of queries by showing document components in some way on the screen. The system allows us to look *into* a document instead of just looking *at* it.

While containing documents, these systems allow a lot of new kinds of queries. For example queries about the semantic content of a document or about all kinds of properties of the *document components*. [3]

But how do we set up such a system? What exactly will be the objects in the database, what structure do they have and how should they be stored? And secondly: how should information be retrieved from them? To answer these questions we have to look at the 'nature' of documents and the way people find information in them.

Logical structure and search strategy

Documents basically have two types of structures: a logical structure (the organisation of document components) and a typographical structure (the way the document is actually presented on paper), both mirroring on a very abstract level the organisation of the semantic content of the document. In hard copy documents only the typographical structure is 'available', the logical structure has to be abstracted by the reader by pattern recognition.[4] For example chapters can be recognized by their highlighted titles, paragraphs by indentation etc. When we are looking for some information in a hard copy document we often make use of the logical structure of the document reflected in its typography.

It is this natural kind of search strategy that we would like to imitate in our retrieval system. An example will illustrate this. Suppose we have an electronic document about the sales of a certain series of computers and we want to know how many computers were actually sold. No system today will be able to retrieve this information directly from the document by *semantic interpretation* of the text (or figures). But if we know or presume that the information we are looking for is contained in some *table*, we can ask the system to look for a document component that has the property of being a *table*. In this way we can shortcut our search actions.[5]

How could we get the system to find a table? More generally, how can the system find document components? By extracting the logical structure from the typography like humans? Recognition of a logical structure on the basis of typography would request computable methods of *pattern recognition*. In addition to the wellknown problems in this field (algorithm complexity and noise, see for example the problems with current OCR systems) The many different kinds of typography used by different authors and editors make any attempt along these lines hopelessly futile. A better and faster way to achieve our goal is to explicitly impose a logical structure on a document.

In what way? It will be clear that for these purposes storage of documents in *plain text format* will not suffice.[6] We will need some *format* to express attributes of the components of a document. If — for example — we want to know the title of a particular chapter of some document, the system will have to know how to recognize a chapter and a title. This can be done by using a markup language.

The idea of imposing a logical structure on electronic documents by using a markup language is not new of course. SGML, ODA and, in a way, LaTeX have been designed for these purposes, although they were primarily intended for improvement of document interchangability (by separation of logical and typographical structure).

So documents need to be stored in two different formats — a logical structure and some typographical represen-

[3] I use the term 'document components' here to indicate the pieces of text (and figures) a document consists of, like paragraphs, sections, footnotes, etc. Document components are pieces of a document that form somehow a logical unit.

[4] And of course by semantic interpretation...

[5] Sometimes the attributes of document components themselves will be the final target of our query. For example we might want to know the exact title of a particular chapter without being interested in its meaning. Or maybe we want to list the particular components of a document that mention Shakespeare, without being interested in the contents of these components.

[6] With *plain text format* I mean a plain ascii format that contains no codes indicating the logical or typographical attributes of document components.

tation — in order to be able to treat queries about logical components that should result in showing typeset pages. If we drop one of them we are either not capable of representing figures or typeset text, or we have to make inferences about the logical structure by pattern recognition (as typeset formats do not contain codes indicating the logical structure information). Moreover if we we want to see documents in a format that *resembles a printed document as closely as possible*, we will probably want them to be sectioned in *pages*.

The role of TEX

If we choose for a pagewise presentation of typeset documents, the two 'faces' of the document now somehow have to be linked by a function assigning pages to document components. This is so because search actions will always take place on the logical face. Whenever a query is meant to get typeset information on the screen, the system will have to know where to find it. Therefore every component in the logical presentation of a document has to contain a *label* — or *adress* — indicating a position in the corresponding typeset representation. In many cases the typeset version is calculated from the logically formatted version on the basis of *font properties* and other graphical information, like grid, size of figures, style, etc.[7] But, as stated before, the typeset version never contains codes indicating the logical structure of a document. So what kind of computer program should take care of the linking? At this point TEX smiles at us.

Because it is the program that does the typesetting that also has to take care of the right pointers from logical document components to physical positions in the typeset version. If we would try to make this preparation module with an ordinary programming language like C or PASCAL we would have to solve typesetting calculation problems that took Donald Knuth ten years while implementing his TEX.

The TEX language and compilers were originally designed to bring professional typesetting within the reach of (scientific) authors. LATEX was meant to combine typesetting facilities of TEX with an advanced strategy of logically structuring documents, as is stated clearly by Leslie Lamport in [7]. I think most of the popularity of both products is (besides — of course — their quality) due to the elegance of this combination.

In my opinion TEX has another great merit: it can be used to elegantly *prepare* documents for databases of intelligent document retrieval systems as described above. The elegance is due to the parallel processing of semantic representations of documents and corresponding DVI-files. More precisely: The TEX compiler can be used to convert a document to a set of two files, one of which is the standard DVI-format, the other one is a file containing all information about the logical structure of a document and about the logical and typographical properties of the document components. But that's not all. The latter file also contains information about the *physical positions* of all components. With physical positions is meant the positions of the DVI-representations of these components in the DVI-file.

In the next section I will sketch briefly a TEX macro set that creates database objects from LATEX documents. This program has been succesfully implemented and can be extended to other types of document (input) formats like WordPerfect or SGML. The program is now incorporated in a prototype document retrieval system that can be combined elegantly with a natural language interface. I need to say at this point that I owe many suggestions and ideas for implementation of the TEX macro set to Huub Mulders, my ex collegue (and the best TEXnician I know) from the Computer Centre of Tilburg University.

A document preparation module in TEX

The main components of the system we designed are a document *preparation module* and a *document retrieval* module. The former is a *format translator*. It is meant to translate formats used by authors (e.g. WordPerfect, WordStar, LATEX) to formats that are convenient for retrieval. The latter is meant to translate queries of a user to operations on a database.[8]

The preparation module is a TEX macro set consisting of a parser and a generator. In the current implementation only LATEX documents can be parsed. From every document (besides the usual DVI-files and auxiliary files[9]) a set of two files is generated: an LST-file and an LSP-file, which will be examined in depth in the next section. For the moment I only want to make the following remarks about them. The former contains the document representation as a nested list of document components and also all text, the latter a set of statements about the sublists of this list (i.e. about the document components). Both files are in plain ascii format and can be directly interpreted by a Prolog interpreter.

Input and output routines

Only TEX's standard input routines are used. To be able to open extra files and write tokens to them I used TEX's `\openout` and `\write` instructions. To split up the information for the DVI-file and for both logical format files I used a recursive macro like:
`\eat#1 {#1 ...\swallow{#1}..... \eat}`.

[7] E.g. LaTeX , Ventura, Interleaf.

[8] At the moment, only a very limited set of query types can be treated.

[9] For those readers not acquinted with TEX: DVI files are typeset images. Auxiliary files are used for several purposes among which calculating logical references like table of contents. A nice introduction in TEX, though in Dutch is to be found in [4]

This macro reads strings from space character to space character. It takes care for both a direct 'consumption' of `#1` by TEX's standard typesetting mechanism and a 'rechewing' `#1` by a (recursive) macro `\swallow`. The latter reads `#1` token by token. Suppose for example `#1` is the string:

(i) `congres.\footnote{...}.This`.

Then `\swallow` makes:
`c o n g r e s \footnote {...} T h i s`
out of it. `\swallow` passes these 13 tokens to another set of macro's that splits up text and logical information.

It will be clear that the `\begin{document}` macro had to be adapted to initiate the `\eat` macro. To properly exit `eat` and other recursive macro's and to account for a proper nesting of document components in the LST file the `\end{document}` macro was also adapted.[10]

These redefinitions are very trivial:

```
\let\begindoc\document%
\def\document#1{%
   ......(instantiations)....\eat#1}%
\let\enddoc\enddocument%
\def\enddocument{%
   .........(handle nestings).....%
            \enddoc}%
```

For the interpretation of tokens routines were needed to check whether two tokens are the same. This is handled by the folowing macro:

```
\newif\ifsame%
\long\def\testsametoken#1#2{%
      \edef\partwo{\string#2}%
      \edef\parone{\string#1}%
      \ifx\parone\partwo\sametrue%
      \else\samefalse%
      \fi}%
```

Because of the token by token reading I needed a macro to append characters to a string. This could be done by the following macro:

```
\newtoks\ta\newtoks\tb%
\long\def\rightappend#1\to#2{%
         \ta={#1}%
         \tb=\expandafter{#2}%
         \edef#2{\the\tb\the\ta}}%
```

Another problem that had to be handled has to do with the linking of the logical and typographical image. Pagenumbers are generated at the beginning of a logical document component. As TEXnicians know TEX does a lot of recalculating at the end of document components. For this reason sometimes a wrong pagenumber was generated. This problem was solved by `\relax`-ing the pagenumber calculation:

```
\def\tolspfile#1{{\let\thepage\relax%
    \xdef\temppar{%
      \write11{\string#1}}}%
    \temppar}%
```

The rest of the macro set comes down to a partly recursive handling of LATEX's component delimiters.

The output format

It is often suggestes that documents should be stored in SGML format ,or a similar markup language type. From a document retrieval point of view this is not very attractive because it implies parsing of SGML documents while treating a query. We cannot avoid 'precompiling' SGML-documents to some other format that allows faster search strategies.[11]

I chose to take a format that has at least two attractive properties. First it is a format that can be *interpreted* by any proper Prolog interpreter and therefore it can also be compiled. Compiling this format results in a low level pointer structure that allows fast searching. Second it is a format that facilitates the interpretation of (natural language) queries. The latter point needs some further explication.

Interpretation of queries

As in any other query language the evaluation of a natural language query comes down to checking relations between objects in a database. In the case of documents and their components these relations are often very particular types of relations that have to do with the *logical positions* of document components, for example the *precedence* relation ($\alpha\ precedes\ \beta$) and the *sublist* relation ($\alpha\ is\_a\_sublist\_of\ \beta$). If we say — for example — that some chapter contains some footnote we say in fact that a list of a certain type is a sublist of another type.[12] What is so special about this? Well, nested lists (like for example *sets*) are in fact *algabraic* entities with a lot of welldefined properties. Basic operations on lists can be implemented in a very trivial way. The idea is then that we can facilitate the evaluation of queries when the database is a nested list and when we define the semantics of our query language in such a way that it can make use

[10] None of LaTeX macro's are redefined in the system files. All redefinitions take place in a distinct top level macro set in a trivial way by temporarily renaming the LaTeX instructions.

[11] A way to avoid parsing SGML bulk files is to make use of SGML's file pointer mechanism. This comes down to putting markups in files that are interpreted as pointers to other files (for details see [1]). In this way sets of files (document components) can be structured in any desired way. In this way we loose an advantage of SGML, namely its direct interchangability with electronic publishing environments.

[12] The term 'type' should not be taken literally here. It is to be discussed whether being a chapter or a footnote is to be taken as a type or as a predicate.

of the basic operations on nested lists when necessary.[13] One might argue that a lot of queries don't request mathematical operations on lists! I think this is indeed the case with queries about the *contents* of a document. A lot of queries however contain subexpressions (for example definite descriptions) that refer to document *components*. Part of the evaluation of the query therefore comes down to finding the correct referents for these subexpressions. Finding a referent on its turn implies an *extraction* of one or more elements from a list, which is in fact a mathematical operation.

EL/DR, a document representation language

EL/DR is a shorthand for *Ensemble Language for Document Representation.* The term 'Ensemble Language' refers to a type of semantic representation languages proposed by Harry Bunt [2]. EL/DR is defined within this framework. The language is used both for document representation and for the semantics of natural language queries. I will only introduce the sublanguage of EL/DR that is used for document representation here. For details about EL/DR the reader be referred to [5].

The language called EL/DR is based on the idea that documents are in fact *tree structures* or *nested lists* of logical components that are themselves also nested lists. For example, a chapter consists of a list of sections. A section consists of a list of subsections or paragraphs and so on. I call those components *DLS* (Document Logical Structure). *Words*, *figures* and (for the moment) *formula's* and *tables* are considered to be atomic. DLS's are syntactically represented in a trivial way. Consider the next (very small) document:

```
Yesterday I bought:
    - two appels
    - a beer
    - milk
    - some bread
```

Suppose we say that this document consists of a paragraph and an enumeration, the latter consisting of four items. This document can now be logically presented as a nested list as follows:[14]

```
[[Yesterday,I,bought],[[two,appels],
      [a,beer],[milk],[some,bread]]]
```

To avoid ambiguities — every word can occur more than once in a document — occurrences of words have to be labelled in a way different from the syntactic representations of the words themselves, for example:

```
[[w1,w2,w3],[[w4,w5],
         [w6,w7],[w8],[w9,w10]]]
```

In fact this is done in EL/DR, but from an implementation point of view it is more convenient to have labels for DLS's (nodes in a tree) too. Therefore, all nodes in a tree structure (atoms included) are implicitly numbered starting from 0 (the whole document) topdown, leftright depth-first, to n, the rightmost atomic element of the tree.

All properties of these DLS's can now very simply be stated in terms of predicate-argument constructions (or for those acquinted with Prolog, in terms of *horn clauses*). For example:

```
paragraph([w1,w2,w3]).
enumeration([[w4,w5],[w6,w7],
             [w8],[w9,w10]]).
item([w4,w5]).
item([w6,w7]).
item([w8]).
item([w9,w10]).
word(w4).
word(w5).
...
```

Or when using the implicitly imposed labels:

```
paragraph(1).
enumeration(5).
item(6).
item(9).
item(12).
item(14).
word(7).
word(8).
...
```

Chapter titles and crossreferences are two-place predicates, for example:

```
title(100,101).
crossref(221,457).
```

The first clause in this example states that the DLS labeled '101' is the title of the DLS labeled '100'. The second clause states that the DLS labeled '221' refers to the DLS labeled '457'.[15]

Finally, the linking of logical components to pagenumbers is done in a similar way, for example:
`pagenumber(221,23).`

The first argument in this example is a label for a logical component, the second argument refers to a pagenumber, which is a physical position in the typeset image.

[13] If we think of representing documents as nested lists we can also make use of their properties in other ways. For example we can use them to *reason* about documents.

[14] For the sake of clarity interpunction is left out.

[15] Note that crossreferences are not seen as attaching *physical positions* in a text to document components, but rather as combining two (or more) logical document components.

Conclusions and final remarks

In this paper I have tried to show that TEX can be used to prepare suitable database objects for document retrieval systems and that alternatives must have particular qualities that are scarcely found in other programming languages. Current software supporting TEX has its limitations though. One of these is that not all graphic information can be represented in DVI-files. This is particularly the case with bitmaps. It seems therefore worthwile investigating alternatives like PostScript.

I have made a lot of presuppositions that are or should be subject of discussion, specifically:

1. advanced document retrieval systems should allow search strategies users have historically got acquinted with (but also — of course — new strategies),
2. hard copies will exist for a long time *together with* their electronic 'sisters'; for this reason a correspondence between the graphic representations of both sisters is desirable,
3. queries that make use of a document's logical structure play an important role in document retrieval.

Still I hope to have made clear that the preparation module for a document retrieval system should output documents as nested lists to make its sublists accessable by mathematical operations in the retrieval module. I also hope to have stimulated the TEX community to think about TEX's hidden qualities in the field of document retrieval.

References

[1] Martin Bryan: *SGML An author's guide to the Standard Generalised Markup Language*, Addison-Wesley Publishing Company, 1988.

[2] Harry Bunt: *Mass Terms and Model-theoretic Semantics*, Cambridge: Cambridge University Press, 1985.

[3] Jeff Conklin: *Hypertext; An introduction and survey*, in: Computer, september 1987.

[4] Victor Eijkhout & Nico Poppelier: *Wat is TEX?*, in: NTG Report 4, november 1989.

[5] Joop van Gent: *A formal language for describing document structures*, forthcoming, 1990.

[6] Donald E. Knuth: *Computers & Typesetting*, Addison-Wesley Publishing Company, 1986.

[7] Leslie Lamport: *LATEX user's guide & reference manual*, Addison-Wesley Publishing Company, 1986.

BIJLAGE O

Towards LaTeX 3.0

Frank Mittelbach and Rainer Schöpf

Abstract

LaTeX is a very valuable tool for document composition. As a TeX macro package, it is unique in its concept of logical commands, at the same time retaining enough flexibility with visually oriented commands to allow the user a relatively easy correction of an automatically chosen layout. This fact makes it far superior to the plain and $\mathcal{AMS}$-TeX macro packages when it comes to professional applications.

Therefore the LaTeX re-implementation project is certainly one of the most important efforts to "expand TeX's horizon". This is the place to link TeX with the modern developments like SGML. This paper describes the current status of the re-implementation of LaTeX.

Objectives of the LaTeX project

LaTeX [6] was developed to serve the specific needs and designs found in documents of natural science, whereas the needs of other fields are more or less neglected. Many layouts and concepts cannot be realized in the present LaTeX, and even those that can be realized are often falsely flagged as "impossible in LaTeX".

One of the main objectives of the LaTeX project is, therefore, to redesign the style file interface by incorporating a broader spectrum of possibilities. At the same time, we try to structure this interface in such a way that it satisfies the needs of a designer. This means that desired layouts should be specifyable preferably through parameters and generic functions that allow a wide range of varieties.

An ensuing objective is the proper documentation of the new interface. It should guide the designer in the evaluation of a new layout, allowing him to use the full power of the interface within a short period of time.

As a third objective, we feel it necessary to re-evaluate LaTeX's internal concepts and reverse those that have been proven inadequate.

In our talks at last year's conferences [8, 9, 10, 11], we presented a concept for a re-implementation of LaTeX. After discussing this topic, Leslie Lamport and one of the authors (FMi) agreed on a two-step procedure, first redesigning the style interface, and then enhancing the user interface.[1]

Further discussion throughout this year has shown that this plan is not feasible as the internal style file interface is affected too greatly by enhancements in the user interface, and vice versa. Therefore, we abandoned this idea and decided to merge both steps, at least temporarily.[2] As a result, the discussion then focused on three different major topics, the enhancements and changes to the user interface, the revision of the style file interface, and the re-evaluation of internal concepts.

In the following sections, we will discuss the topics which we have been concerned with since last year's conference and the state of their realisation.

The User Interface

Any change to the user interface of an existing program always opens the question of compatibility. While we judge this question as very important, we feel that it is not justified, for the sake of upward-compatibility, to leave all existing features untouched, even when they have proven to be inadequate. This means that we try to keep this sort of changes small, and devise a possibility to emulate LaTeX 2.09 in the new LaTeX to allow processing of older documents with no changes or only a few.

Attribute concept

It has been generally agreed in the ongoing discussion that an attribute concept as supported by DCF GML [4, 5] and SGML [3] would be a great improvement. Since this is a major change, it was discussed whether such a concept would render the optional arguments obsolete.

Reprinted from TUGboat 12 (1991), No. 1 – TeX90 Conference Proceedings © 1990, TeX Users Group; reprinted with permission.

[1] This was published as the update section in [8] and [7].

[2] We feel that it is still sensible to defer certain parts of the revision to a later stage. However, this will only concern internals of the implementation and not induce any changes to the user or the style-designer interface.

But as the number of LaTeX users is far greater than most people think, such an incompatible change in the syntax is not feasible. In additon, the old optional arguments and star forms provide convenient short forms for the most important attributes. While there have been several proposals for an attribute syntax as well as one prototype implementation, a final decision has not been made as yet. But it seems probable that this concept will only be available for environments, not for ordinary commands. We think that this will be sufficient since it is planned to provide environment forms of all text producing commands (cf.).

Robust and fragile commands

The distinction between robust and fragile commands will no longer be present. Instead, all text in moving arguments will be automatically protected against expansion.[3]

To allow the style file writer the specification of text that has to be expanded, there will be a command that removes or partly removes this protection.

We do not consider it necessary to give this feature to the user. Hence, this is not available in the user interface.[4]

Font selection

The new font selection scheme is already being distributed as beta test version for LaTeX 2.09 and seems to be working very well. It is also the basis for the `amsfonts` style option that makes the $\mathcal{AMS}$FONTS collection available for LaTeX and is part of the $\mathcal{AMS}$-LaTeX distribution [1].

Nevertheless, the current implementation should not yet be regarded as the final product. Time and users' demands will show whether it has to be improved.

Front matter

The specification of preliminary material is one of the parts which are not handled properly in LaTeX 2.09. Although this topic has not been discussed in depth so far, this might be one of the places where the syntax of the new LaTeX might differ in an incompatible way.

Tables

The extensions of the `array` and `tabular` environments by Frank Mittelbach [12] seem to be widely accepted. Several further extensions are conceivable, but this needs careful evaluation. There is also a new implementation of these environments by Denys Duchier that includes the extended syntax. This implementation looks very promising and can probably serve a basis for table handling in the new LaTeX.

We will provide a command to specify notes to tables working similar to the `\footnote` command inside a `minipage` environment.[5]

Math

The `amstex` style option for LaTeX 2.09 implements most of the features of $\mathcal{AMS}$-TeX in LaTeX syntax (such as `\begin{align}`...`\end{align}` instead of `\align`...`\endalign`). As this is now being distributed by the AMS, users are able to typeset complicated math formulas in LaTeX without falling back to plain TeX's idiosyncrasies [1].

However, the implementation still has some loopholes that need to be eliminated in future versions.[6]

Text producing arguments

All commands with arguments in which the user specifies text to be typeset (e.g., `\fbox`) will also be available in an environment form to allow their use in user-defined environments.

Verbatim input

The use of the `\verb` command will be possible in all circumstances.[7] A similar extension of the `verbatim` environment is not possible.[8] But this restriction is lessened by the possibility to use the environment form of the respective command in which the `verbatim` environment may be used.

Float positioning

LaTeX 2.09 was designed for documents containing only relatively few floats.[9] The new implementation will improve the float position algorithm and the user's control.

[3] The approach of LaTeX 2.09 to expand everything by default is counter-intuitive and a common source of nasty errors.

[4] Expansion is normally necessary to cope with problems presented by the asynchronous output routine mechanism together with macros that change their contents, so that it is essential to write the expansion and not the macro name to a file, etc.

[5] Using `\footnotemark` and `\footnotetext` commands inside a table that (additionally) has to be put inside a `minipage` environment is another counter-intuitive concept of LaTeX 2.09.

[6] Some features do not work correctly in boundary cases. This is partly due to limitations in the current LaTeX, e.g., the primitive handling of `\begin`...`\end` (see below), etc.

[7] However, due to limitations of the TeX program itself, the use of multiple blanks in one `\verb` command will not be supported in all cases.

[8] In LaTeX 2.09 neither the `\verb` command nor the `verbatim` environment may be used in arguments.

[9] If, for example, the space for floats is larger than the surrounding main text, as is often the case in appendices of manuals, etc, LaTeX 2.09 is seldom able to compile the document without running out of memory space, even if the float parameters are given full flexibility.

This might include some sort of an 'h' option that really means "here".[10] There have also been proposals to prohibit the use of multiple captions within one float, thus allowing the document style to position the caption according to its own rules. But these topics have not yet been discussed thoroughly enough to present a final concept.

Bibliographies

The handling of citations and bibliographies will probably change to support several conventions. Since this topic depends on the development of the new BibTeX to some extent, it is not yet clear what is actually implementable.

Specific problems concerning citations and the interaction with BibTeX are discussed in [14] and in [13].

Omitting environment end tags

The implementation of a prototype for error recovery in case of unmatched `\end` tags has shown that it is possible to implement the concept of implied `\end` tags. This feature will help in writing SGML parsers that produce LaTeX output. Whether the detection of implied `\begin` tags, e.g., the omission of the first `\item` in a list environment, can be easily implemented requires further testing.

The Style-Designer Interface

As we stated above, the main goal for the design of the style file interface is making it easily applicable. Therefore, the new interface will contain a lot more generic commands allowing for the specification of a wide range of layouts with a minimum of effort.

International language support

Support for more than one language (US English) was one of the key issues that triggered this LaTeX reimplementation project. In the new implementation, all textual representations in style files will be settable. While this is certainly not sufficient to support the different typographic conventions of different countries, it allows for typesetting foreign texts within the usual typographic conventions.

Hooks

For the implementation of certain layouts, it is often necessary to carry out specific actions at well-defined points, e.g., the footnote placement algorithm of this article has to be initialized at `\begin{document}`. For this type of applications, many of the internal commands will contain hooks that allow the style file writer to add code to these commands without overwriting the original definitions.[11]

Generic section headers

One goal for the style file interface is to provide the designer with a generic heading macro which implements a broad range of layouts by varying certain parameters. It is clear that a proper balance has to be found between the internal complexity of such a command and the number of different layouts which are specifyable through it. Several different syntax proposals have been discussed so far, but the discussion hasn't reached a satisfactuary conclusion, as yet.

The new mechanism will probably provide a specification for headings, in which the designer has complete control over heading layout, e.g., positioning supplied text[12], ornaments[13] and the like. It is planned to support the following general types of headings:

- Vertically oriented headings, where the heading is separated by white space from preceeding and following text.[14] There will be parameters to allow the heading to extend into one or both margins.
- Horizontally oriented headings, where the heading is partially or fully placed into one of the margins beside the following text.[15] A critical problem with this sort of layout is to guarantee that enough space is available and that any necessary hanging indention (for headings placed only partly into the margin) is applied up to the necessary point.
- Run-in headings, where the text following continues on the same line as the heading.[16]

The designer will be given tools for specifying the heading layout in an easy manner, so that it is possible to vary the layout depending on things like the length of the heading text, the presence or absence of a heading number, etc.

It is planned to allow for the specification of a minimal amount of text that has to follow a heading.[17]

[10] We are aware of the problem of handling previously deferred floats of the same kind.

[11] In LaTeX 2.09 a lot of style options are incompatible with each other, simply because they redefine the same internal macro.

[12] For example, 'Chapter' in the `\chapter` command.

[13] Rules and dingbats, etc.

[14] This layout has already been realized to a certain extent in the `\@startsection` command of LaTeX 2.09. But this generic macro is not able to specify the layout of the heading (except setting the used font), so that it can not even be used for standard headings like the `\chapter` command.

[15] This sort of layout is not supported in the current version.

[16] Again, this layout is provided in LaTeX 2.09 but does not allow specifying the layout of the heading, e.g., punctuation marks at the end, underlining, etc.

[17] In LaTeX 2.09 this is the fixed amount of two lines of text. It might be possible that a more general implementation will fail

Generic table-of-contents entries

What has been said in the last subsection about generic section headers applies to the formatting of entries in the table of contents to as well. This has not yet been discussed in detail, but we hope that a proper implementation of generic section headers can be used as a starting point for generic toc entry formatting. The same mechanism can then be used for other types of tables as well (`lof`, `lot`, etc.).

Generic lists

Necessary extensions and corrections of the generic list environment have been already discussed in [8, 11]. There have been a few proposals concerning lists, but this topic needs further attention, as it affects a major part of most style files.

Parameter tables

There has been a proposal to group certain parameters into tables to make their structure and dependence visible. One item that will probably change on the style designer level in this way is the specification of default parameters for different levels of lists. But it is not yet clear, whether such a concept is implementable in an efficient manner.

Paragraph design

The specification of paragraph layout (such as different forms of ragged right typesetting) will be improved. The designer will be given the possibility to specify such layouts not only for the main text, but also for footnotes, floats, etc.[18]

Toc levels

There are book designs which require several tables of contents, e.g., one for the whole book and those refering to each chapter. This touches on the topic of auxiliary file handling (cf.) since the current mechanism does not allow more than one table per document. There will be support to select only certain entries of a table of contents for printing. A prototype implementation for this feature was written by Nico Poppelier.

Documentation

The interface between the style files and the LaTeX kernel will be documented properly. We will provide a complete description as well as a number of examples.

Internals

Error recovery

In the current LaTeX, an omitted or misspelled environment name usually produced a lot of error messages and often supressed any further compilation of the document. In the new implementation, certain classes of environment errors are detected and corrected without damaging the output. For this complex, a prototype implementation is currently undergoing alpha testing. It implements the following features:

Incorrect `\begin` tag If the user misspells the name of the environment desired inside the `\begin` tag, an error message is generated and the offending `\begin` tag is ignored.[19] However, the user is allowed to insert the correct environment name by specifying `i\begin{`⟨*envir*⟩`}` in response to the error message.[20]

Incorrect `\end` tag When an incorrect `\end` tag is encountered, e.g., `\begin{bar}` ... `\end{foo}`, the new LaTeX tries the following recovery:

1. If `\end{foo}` is unknown, we assume that the user misspelled the name and recover by replacing `\end{foo}` with `\end{`⟨*curenvir*⟩`}`. This will produce an error message but will allow safe continuation of the compilation afterwards.
2. If `\end{foo}` is a legitimate `\end` tag, i.e., if the corresponding internal environment start command is defined, we check to see whether there are any unresolved `\begin{foo}` environments.
   a. If so, the currently open environment is closed by inserting an `\end{`⟨*curenvir*⟩`}` tag.[21] Afterwards `\end{foo}` is tried again. This mechanism will close all open enviroments until the correct one is found. Depending on the status of an internal variable, this will either produce an error message, or a warning message, or will be executed silently to allow for the implementation of implied `\end` tags.
   b. If there is no open `\begin{foo}` we simply ignore the `\end` tag after issuing an appropriate error message. The underlying idea is that this `\end` tag is probably left over after moving some text in the source around.

in special cases due to TeX limitations.

[18] In LaTeX 2.09, this is not possible without redefining several internal macros, because the paragraph shape parameters are reset at several points to fixed defaults.

[19] Of course, this mechanism will be triggered only if the user did not exchange one environment name for another.

[20] In LaTeX 2.09, this response would result in TeX error messages at the end of the compilation.

[21] The implementation of this part of the recovery is not as straight forward as it may seem. The term ⟨*curenvir*⟩ does not necessarily refer to the innermost open environment because it may be possible that a user defined enviroment calls other enviroments in its body. In such a case, the calling environment has to be closed first, because the inner environments are resolved in its `\end` tag.

Auxiliary file handling

We will implement a two-step approach for `.aux` files where all information is written to an intermediate file which is then copied to the 'real' `.aux` file at the end of the run. As a result, there will be only one `.aux` file instead of the many files produced by LaTeX 2.09 when the `\include` command is used. The new scheme will make it possible to preserve cross-reference information if a compile run ended prematurely.[22] In addition, it will be possible to detect whether `\include` files have to be re-compiled because of changes in other parts of the document, or not.

References

The use of symbolic references will be extended to include textual references, if desired.[23] Whether this will result in some changes concerning the user interfaceor not has not been discussed so far. If it is feasible, we will also provide the possibility for hierachical references.[24]

Page selection

To provide easy access to the different parts of a document at the printing level, we plan to record certain document structure information in the `\count` registers 0–9. Besides the usual page counter in `\count` 0, this might include the physical page number, the current chapter, section, or subsection, ... number, to allow the printing of, e.g., Chapter 6 or "all preliminary pages"[25] by giving a simple page selection pattern to the printer driver.[26]

Plain TeX compatibility

We tend to build up the new LaTeX from scratch, i.e, not to read in the plain TeX format (or a nearly identical variant) as a basis when building a format file.[27] This does not mean the useful functions, `\mathchar` definitions, etc. of `plain.tex` will be discarded, but concepts which are obsolete in the LaTeX environment or macros that can be implemented in a better way will be replaced or removed.

Beta testing

It took LaTeX about three years to develop into a stable system. To avoid needing a similar period of time when switching to the new version, we plan to run the new version throughout the development at a few selected sites for beta testing. So, if you are a maintainer of a TeX installation and think that you can persuade your users to play willing (or unwilling) guinea pigs, please, contact one of the authors. Your installation should have the following characteristics:

- A running TeX 3.0 preferably (but not necessary) with drivers supporting the virtual fonts introduced with TeX 3.0.
- A fair amount of LaTeX processing in a fully supported LaTeX 2.09 environment (including a customized Local Guide) and at least one user with some experience in style writing.
- A working eMail connection.
- A maintainer (you) who is willing to
  - provide backups and fast user support in case something goes wrong
  - send in bug reports, if necessary
  - compile new formats when updates arrive.

The work that has to be carried out by the beta testers should not be underestimated. It is probably a time-consuming commitment since a lot of organizing is usually involved. Nevertheless, we hope that there will be enough people around willing to help us in this stage of development, so that we can finally return a product to the user that will have the same success as the current LaTeX had.

Throughout the beta testing phase, LaTeX will have to show its abilities in real life situations. As TeX and LaTeX are currently finding their way into new areas, we hope that people from these fields will take the opportunity to test whether the new LaTeX matches their needs by participating in the testing phase. This is the time when it is still possible to correct errors before they become established as unfortunate facts.

References

[1] American Mathematical Society, Providence. *$\mathcal{A}_{\mathcal{M}}S$-LaTeX Version 1.0 User's Guide*, July 1990.

[2] Brüggemann-Klein, Anne, editor. *1989 EuroTeX Conference Proceedings*, 1990. To appear.

[22] The current implementation writes directly to the `.aux` file, so that its contents are damaged or lost when an error occurs.

[23] In the current LaTeX, references can only be made to counter values, or more exactly, to a combination of counter values. This will lead to problems if document styles decide to use unnumbered headings, for example.

[24] In LaTeX 2.09, this has been realized for the `enumerate` environment, to some extent.

[25] In LaTeX 2.09, it is often not possible to print a specific set of pages that have the same TeX page numbers as other pages that come earlier in the document.

[26] With most printer drivers, the first page to print can be selected by specifying a pattern to be matched against TeX's `\count` registers 0–9. We propose that drivers also allow for the specification of a pattern for the pages to be included.

[27] Currently, LaTeX is built on top of `lplain.tex` which differs from `plain.tex` in only a few places. This means that even the funny keyboard support (for SAIL terminals) is included that will give you in certain situations a '⊕' character if you key in `$^^M$`.

[3] Bryan, Martin. *SGML: an author's guide to the standard generalized markup language*. Addison-Wesley, Woking, England; Reading Massachusetts, second edition, 1988.

[4] *DocumentComposition Facility: Generalized Markup Language Starter Set User's Guide*. New York, fifth edition, May 1987.

[5] *Composing Documents with the Generalized Markup Language*. International Business Machines Corporation, New York, second edition, March 1988.

[6] Lamport, Leslie. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 1986.

[7] Mittelbach, Frank and Rainer Schöpf . "A new font selection scheme for TeX macro packages—the basic macros." *TUGboat*, 10(2):222–238, July 1989.

[8] Mittelbach, Frank and Rainer Schöpf . "With LaTeX into the nineties." In Thiele, Christina, editor, *1989 Conference Proceedings*, volume 10#4 of TUGboat, pages 681–690. TeX Users Group, December 1989.

[9] Mittelbach, Frank and Rainer Schöpf . "LaTeX dans les années 90." *Cahiers GUTenberg*, (6):2–14, July 1990. French translation of [11] and of some parts of [10].

[10] Mittelbach, Frank and Rainer Schöpf . "LaTeX limitations and how to get around them." In *1989 EuroTeX Conference Proceedings*, Anne Brüggemann-Klein [2].

[11] Mittelbach, Frank and Rainer Schöpf . "With LaTeX into the nineties." In *1989 EuroTeX Conference Proceedings*, Anne Brüggemann-Klein [2].

[12] Mittelbach , Frank. "A new implementation of the array–and tabular–environments." *TUGboat*, 9(3):298–314, December 1988.

[13] Read, David. "Towards BibTeX style-files that implement principal standards." *TeXline*, (10):2–8, May 1900.

[14] Wonneberger, Reinhard and Frank Mittelbach . "BibTeX reconsidered." In Guenther, Mary, editor, *TeX 90 Conference Proceedings*, page ?, ? 1990. Cork September 1990, to appear.

Getting TEXnical: Insights into TEX Macro Writing Techniques

Amy Hendrickson
TEXnology Inc., 57 Longwood Avenue, Brookline, MA 02146
617-738-8029 Internet: amyh@ai.mit.edu

Abstract

Most of us understand the basic form of TEX macros but that understanding alone is often inadequate when we need to solve certain problems. We need additional insight to be able to develop methods of passing information, moving text with changed catcodes, preserving blank lines, and more. Writing a large macro package brings in a new set of issues: how to avoid bumping into implementation restrictions, e.g., constraints of hash size, string size, and others; how to make a pleasant user interface; how to make your code as concise as possible.

Some of the techniques to be discussed here include making a macro with a variable number of arguments; changing catcodes in macros, defining a macro whose argument is intentionally never used; conserving hash size by using counters instead of newifs; csname techniques and non-outer dynamic allocation; and table making techniques. Finally, some suggestions are included on methods to use when developing new macros.

A Quick Review of Some Important TEX Primitives

Expandafter. `\expandafter`, a TEX primitive often used in this article, affects the timing of macro expansion. Macro expansion is that step in TEX's processing which changes a control sequence to whatever that control sequence is defined to represent. `\expandafter` is usually followed by calls to two macros. It expands the *first* macro following it only after it has expanded the *second*. Thus, `\expandafter` makes it possible for the first macro to process the pieces of the second macro as if the second macro were written out, not represented by a control sequence.

Here is an example: If we define `\letters` and `\lookatletters`,

```
\def\letters{xyz}
\def\lookatletters#1#2#3{First arg=#1,
  Second arg=#2, Third arg=#3 }
```

and follow `\lookatletters` with `\letters ? !`, `\lookatletters` takes the whole definition of `\letters` as the first argument, `?` as the second argument, and `!` as the third. Thus

```
\lookatletters\letters ? !
```

produces

First arg=xyz, Second arg=?, Third arg=!

But if we use `\expandafter`, `\lookatletters` will be able to process the contents of `\letters` for each argument:

```
\expandafter\lookatletters\letters ? !
```

produces

First arg=x, Second arg=y, Third arg=z ? !

String. `\string` is a TEX primitive which causes the control sequence following it to be broken into a list of character tokens in order to print the control sequence or to process it with another macro. `\tt\string\TeX` will produce `\TeX`. (What is the `\tt` doing in there? It makes the backslash print as backslash (\) when it would otherwise print as a quote mark ("). If you are curious about this, look up `\escapechar` in *The TEXbook*.)

Csname. `\csname ...\endcsname` is an alternative way to define and invoke a TEX command. Its function is the inverse of that of `\string`. `\string` takes a control sequence and turns it into tokens; `\csname...\endcsname` takes tokens and turns them into a control sequence.

Commands called for with `\csname` produce the same results as the backslash form (e.g., `\csname TeX\endcsname` and `\TeX` are equivalent) but the `\csname` construction combined with `\expandafter` allows you to build and invoke a control sequence dynamically at the time the file is processed, as opposed to knowing its name at the time the macros are written. This technique has many interesting and useful applications, as we will soon see.

If and ifx. Since both `\if` and `\ifx` are conditionals used to compare tokens, the TEX user may well wonder when to use `\if` and when to use `\ifx`. When we understand how each of these conditionals works, we may conclude that the answer is to use `\if` *only when comparing single tokens* and to use it with care.

How 'if' works. `\if` expands whatever immediately follows it until it arrives at two unexpandable tokens. It then compares them to see if their charcodes match. This test is useful to see if a given letter is upper- or lower-case and in some other instances where we need to test a single token.

The two tokens that are compared are the first that appear after `\if`, *even if they are both* found inside the same macro following it. Understanding that principle makes sense of these samples which would otherwise be mystifying.

```
\def\aa{ab}
\def\bb{ab}
\if\aa\bb
```

tests false, because `\if` expands `\aa` and compares 'a' with 'b'. Whereas

```
\def\aa{aa}
\def\bb{bb}
\if\aa\bb
```

tests true, because TEX compares 'a' with 'a' in the macro `\aa`. `\if` doesn't process `\bb` since it has already found two unexpandable tokens and in this case will cause the letters 'bb' to print since the conditional is set to true and `\bb` is found in the true part of the conditional.

There is another problem to consider. Since `\if` expands a control sequence to its bottom level, meaning every control sequence that is found in the definition of a command being expanded will itself also be expanded, it may generate an error message if a control sequence is expanded that contains an `@` in its name.

This problem arises because Plain TEX commonly includes `@` as part of macro names, with the catcode of `@` set to that of a letter. The catcode of `@` is set to 'other' in normal text so that when a Plain TEX command of this sort is expanded in text the `@` is no longer understood as a letter, and TEX will give the user an error message about an undefined control sequence. For example,

```
\if\footnote X  Yes\else No\fi
```

produces this error message:

```
! Undefined control sequence.
\footnote #1->\let \@sf
                        \empty \ifhmode...
```

How 'ifx' works. `\ifx`, on the other hand, will not have this problem since it only expands to the first level of macro expansion. If `\dog` is defined by `\def\dog{\cat}`, for instance, `\ifx` will expand `\dog` as far as `\cat` but will not expand `\cat` to use its definition.

This means that when we want to compare control sequences, and to supply one control sequence as an argument to a macro, we can use the `\ifx` conditional to look at the name of the macro supplied without having to worry about macros that may be contained in its definition.

For example, we can define `\def\aster{*}` so that we can use it to compare with another macro. Inside the macro where we want to make the comparison, we can write

```
\def\sample#1{\def\one{#1}\ifx\one\aster...
```

making both the argument to `\sample` and * be defined as macros.

When `\sample` is used, `\ifx` causes only one level of expansion. If the argument given to `\sample` is `\footnote`, as in the `\if` example above, `\one` will be defined as `\def\one{\footnote}`. `\ifx` will expand `\one` to find '`\footnote`' but will not expand it any further, and will not give an error message.

Another reason to use `\ifx` to compare control sequences is that `\ifx` will pick up both control sequences following it and compare them. When we try the same samples with `\ifx` that we did with `\if`, we will find that we get results opposite to those we got with `\if` — and we will get the results we would want when comparing control sequences.

```
\def\aa{ab}
\def\bb{ab}
\ifx\aa\bb
```

tests true, because `\aa` and `\bb` match each other in their first level of expansion, whereas

```
\def\aa{aa}
\def\bb{bb}
\ifx\aa\bb
```

tests false because the first level of expansion of `\aa` and `\bb` do not match.

Picking Up Information

Defining a macro that will pick up and process a variable number of arguments. There are many instances where you might want to allow a variable number of arguments. Table macros are one such case, in which the user might supply the width of each column as an argument, and the number of columns may well vary from table to table. A table alignment macro that determines whether each column in the table should be aligned to the right, left, or center, is another case where processing a variable number of arguments is necessary.

There is a general method for constructing a macro that will accomodate a variable number of arguments. This method is to pick up all the arguments as one unit and then take that unit apart as a second step. For example, `\table 1in 2.3in 4in*` can be the command to start a table using dimensions to specify the width of each column. When `\table` is defined as `\def\table#1*{....}` we can pick up all the dimensions as the first argument, since the first argument ends with `*`, then use a second macro to process each dimension as its argument. The second macro will call itself again after each dimension is processed until all the dimensions have been used.

Here, in a sample macro, we define `\pickup` as `\def\pickup#1*{...}` to use it to pick up everything between `\pickup` and the `*` as its first argument. Then we use `\expandafter` to allow `\lookatarg` to process the contents of the first argument.

```
\def\pickup#1*{\expandafter\lookatarg#1*}
```

The definition of `\lookatarg` contains a looping mechanism: It is a conditional that tests to see if its argument is equal to '`*`'. It will keep calling itself (recursing) until its argument is `*`. It calls itself by redefining the command `\go` within the conditional, and calling for `\go` outside the conditional. (`\go` must be placed outside the conditional. If it were to be used inside the conditional it would take the `\else` or the `\fi` as its argument and massive confusion would result.) When `\lookatarg` sees '`*`' as the argument, it will define `\go` as `\relax` and thus will not call itself again.

First we define `\aster` so that we have a command to use with `\ifx` to compare with the argument of `\lookatarg`:

```
\def\aster{*}
```

Now we can compare the argument of `\lookatarg` with `\aster`. Thus, with

```
\def\lookatarg#1{\def\one{#1}
  \ifx\one\aster\let\go\relax
  \else Do Something \let\go\lookatarg
  \fi\go}
```

if we use the `\pickup` macro as follows

```
\pickup abc def*
```

the results would be:

Do Something Do Something Do Something Do Something Do Something Do Something

`\lookatarg` has been invoked 6 times since it picked up 6 tokens before it found the `*`. We can substitute some other command for 'Do Something' and build a more useful macro.

Here are two applications of the technique demonstrated in `\lookatarg`; a macro to underline every word in a given section of text, and a macro to process a given section of text to imitate the small caps font.

First, we define `\underlinewords`, which picks up the whole body of text to be underlined:

```
\long\def\underlinewords #1*{%
  \def\wstuff{#1 }\leavevmode
  \expandafter\ulword\wstuff * }
```

Here `\leavevmode` asks TEX to go into horizontal mode. Since each word will be placed in a box, we need this command to prevent the boxes from stacking vertically, as they would in vertical mode.

Now we define `\ulword` which will unpack the text picked up, word by word, put each word in a box, and provide a horizontal rule under each:

```
\long\def\ulword#1 {\def\one{#1}%
  \ifx\one\aster\let\go\relax
  \else\vtop{\hbox{\strut#1}\hrule \relax}
    \let\go\ulword
  \fi\go}
```

The space given after the argument number in the parameter field will allow us to pick up one word at a time, since the collection of the argument will be completed only when `\ulword` sees a space. Here we use `\underlinewords`:

```
\underlinewords
non-outer dynamic allocation*
```

which results in:

non-outer dynamic allocation

The macro `\fakesc` is another construction using this technique. It lets you set text in large and small caps, imitating the 'small caps' font. Its arguments are, in order, the font for the larger letters, the font for the smaller letters, and the text that is to be set in small caps.

When we use `\fakesc` we need to declare the two fonts to be used:

```
\font\big=cmr10
\font\med=cmr8
```

and then

```
\fakesc\big\med  Here are Some Words to be
Small Capped. NASA, Numbers, 1990*
```

will result in:

HERE ARE SOME WORDS TO BE SMALL CAPPED. NASA, NUMBERS, 1990

The macro starts by defining the two fonts and the text to be processed; there is a space after `#3` in `\def\stuff{#3 }` because `\pickupnewword` needs a space to complete its argument when the last word is found as `\stuff` is expanded:

```
\def\fakesc#1#2#3*{\def\bigscfont{#1}%
  \def\smscfont{#2}\def\stuff{#3 }%
  \expandafter\pickupnewword\stuff *}
```

`\pickupnewword` picks up one word at a time, in order to preserve the space between words. If we just asked `\pickupnewlett` to process the entire third argument of `\fakesc`, the space between words would be thrown away as irrelevant space appearing before the next character being looked for as the argument of `\pickupnewlett`. Here, then, is the definition of `\pickupnewword`:

```
\long\def\pickupnewword#1 {%
  \expandafter\pickupnewlett#1\relax}
```

Once `\pickupnewword` has picked up the word, `\pickupnewlett` is used to test each letter to determine whether it should be capitalized. If so, it uses the larger size font; otherwise, the smaller. `\pickupnewlett` tests to see if the argument is uppercase by using the first argument to define `\letter`, `\def\letter{#1}`, and then defines `\ucletter` in an uppercase environment.

```
\uppercase{\def\ucletter{#1}....}
```

Now it uses the `\if` conditional to compare `\letter` and `\ucletter` If they match `\pickupnewlett` makes the current letter or number be printed in the larger font; otherwise the smaller font is used. Note that we can use the `\if` conditional here since we are only comparing single letters. So, finally, the definition of `\pickupnewlett`:

```
\def\pickupnewlett#1{\def\letter{#1}%
  \if\letter*\unskip\let\go\relax
  \else%
    \if\letter\relax{\bigscfont\ }%
      \let\go\pickupnewword
    \else\uppercase{\def\ucletter{#1}%
      \if\letter\ucletter%
        {\bigscfont#1}\else{\smscfont#1}
      \fi}%
      \let\go\pickupnewlett
    \fi\fi\go}
```

When to pick up text as an argument, and when to to pick up text in a box. The correct timing of catcode changes is an issue of concern to the macro writer. Picking up text as an argument will usually be the right way to provide information for the macro, but will fail if you need to change catcodes, since catcodes are irrevocably assigned at the time TEX reads each character. Thus, by the time TEX has picked up an argument, the catcode of all the tokens in the argument are set, and no amount of fiddling with the argument within a macro will change this.

Even a catcode change asked for in the body of an argument will not effect a catcode change because the catcodes of the tokens will already be set by the time TEX expands the request for the catcode change.

There are two ways to solve this problem. In simple cases, one can build a macro containing the desired catcode changes and then invoke a second macro within the first, i.e.,

```
\def\changecat{\bgroup
  \obeylines\pickupchanged}
\def\pickupchanged#1\endchange{%
  \setbox0=\vtop{\hsize=1in#1}%
  \centerline{xx\vtop{\unvbox0}yy}%
  \egroup}
```

In `\changecat` a catcode change is produced by `\obeylines` which changes the catcode of the end-of-line character, `^^M`, to 13 ('active') so that it can be defined as `\par`. Once that catcode change is made, `\pickupchanged` is invoked. Its argument has the end-of-line character set to category 13 at the time the argument is picked up. Notice the `\bgroup` command in `\changecat` is matched with the `\egroup` command in `\pickupchanged` to confine the catcode change.

Used:

```
\changecat
What
Happens
Here?
\endchange
```

xx What yy
Happens
Here?

But, though this example will work for a catcode change set within the `\changecat` macro it will not allow `\changecat ...\endchange` to pick up an argument that contains a catcode change, for instance, an argument containing macros to produce verbatim text, as we could do with the following technique.

The two-part macro defined below will build a box, starting it in `\pickupcat` with `\setbox0\vtop \bgroup`. Any material found between it and `\endpickup` will be expanded, and finally the box will be

completed with the \egroup found in \endpickup, a construction that allows a catcode change between the first and second part of the macro.

```
\def\pickupcat{\global\setbox0
  \vtop\bgroup\hsize=1in\obeylines}
\def\endpickup{\egroup%
  \centerline{XXX\vtop{\unvbox0}YYY}}
```

\pickupcat...\endpickup is shown using a previously defined verbatim environment, \beginverb...\endverb, to pick up and move verbatim text in a box. Once we see this principle, we can see that a revision of this macro would allow us to place verbatim text in a figure environment, or in a table, or in another environment where it would normally be difficult to introduce material with changed catcodes. For one example:

```
\pickupcat
\beginverb
   Test of
%$_^#@\
    Verbatim

  text.
\endverb
\endpickup
```

produces:

```
XXX   Test of    YYY
    %$_^#@\
        Verbatim

      text.
```

Looking ahead at end of line to preserve blank lines. Since TEX normally ignores blank lines between paragraphs and in some cases we might want to maintain blank lines, we need to develop a way to test for blank lines and provide vertical space when one is present. In this case, we are not interested in picking up text but in picking up information. What comes after each end-of-line character?

As previously mentioned, \obeylines changes the end-of-line character, ^^M to \par. You can define ^^M to do other things as well. For instance, you can define it to be a macro that will supply a baselineskip when the next line is blank or a lineskip when the next line is not.

In this example, ^^M will be defined as \lineending, a macro that includes \futurelet to look ahead in the text. If the character that it sees is *itself* (\lineending), the next line is blank, since there is nothing from one end-of-line character to the next one. The macro \looker will then supply a baselineskip. If it does not see itself, indicating that the next line is not blank, \looker will supply a lineskip:

```
\def\looker{%
  \ifx\next\lineending%
    \vskip\baselineskip\obeyspaces\noindent%
  \else%
    \ifx\next\endgroup\else%
      \vskip\lineskip\obeyspaces\noindent%
  \fi\fi}

\def\lineending{\futurelet\next\looker}

{\obeylines
\gdef\saveblanklines{\bgroup\obeylines%
  \let^^M=\lineending}}

\def\endsavelines{\egroup}
```

Example:

```
\saveblanklines
Here is

a blank line,
and a non-blank line.
\endsavelines
```

which produces

Here is

a blank line,
and a non-blank line.

Passing Information: When Counters Can be More Advantageous than Newif's

Hash size, the size of that part of TEX's memory in which it stores control sequence names, is usually not something about which the macro writer has to be concerned. When building a large macro package, however, hash size can be exceeded, making the number of control sequences defined an important issue. One way to economize on the number of definitions in a package is to use counters to pass information rather than using \newifs.

When the number of control sequences is not important, \newif can be used to create a conditional. This conditional can then be set to true or false in one macro, and tested to see if it is true or false in another as a way of passing information from one macro to another.

However, every time a \newif declaration is used, three new definitions are generated. If saving hash size is an issue, we can use \newcount instead, and only one new definition is generated.

We can use \newcount to allocate a counter and assign it a name, e.g., \newcount\testcounter. Then, instead of setting a conditional to true or false, i.e., \global\titletrue, and testing for it, i.e., \iftitle ...\else ...\fi we can test for the value of the counter. For example,

```
\global\testcounter=1
```

could be the equivalent of `\global\titletrue`. We can then use this test:

```
\ifnum\testcounter=1 ...\else ...\fi
```

Counters have the additional advantage of allowing you to test for a range of numbers, i.e.,

```
\ifnum\testcounter>1 ...\else ...\fi
```

so you can write more compact code when testing for a number of options.

For instance, if we were to write a macro that allowed the user to choose to fill a box by pushing the text to the left, center, or right, we could assign a numerical value to each of the options. If we assigned a value to `\testnum` according to the plan, `left=0, center=1, right=2`, we could test for a range of numbers when another macro was determining which way to fill the box. The test could look like this:

```
\hbox to\hsize{\ifnum\testnum<1
    %% if text is to be pushed to the left
\else
    %% if text is to be either centered or
    %% pushed to the right, do \hfill
  \hfill\fi
  ⟨text⟩
\ifnum\testnum>1
    %% if text is to be pushed to the
    %% right, don't do \hfill
\else
    %% or text is either centered
    %% or pushed to the left
  \hfill\fi}
```

This same principle can be used in more complicated cases as a way of reducing great masses of nested conditionals to a test of the range of the value of a particular counter.

Methods of Conserving Hash Space

As mentioned earlier, using counters to pass information rather than `\newif\thinspace` s is one way to help prevent the hash size from being exceeded. Here are some others.

Input separate macro files on demand. To reduce the number of macros in a macro file, break up the complete macro package into a general macro file and a number of secondary macro files. Within the general macro file definitions can be made that read in the secondary files only when the user calls for a macro for a particular function. For instance, a file containing all the table macros will only be read in if the user uses the general table macro. This principle can be used for listing macros, indexing macros, and any other sort of macro that will not necessarily be used for every document.

Using non-outer dynamic allocation. Dynamic allocation is the way macro writers are able to access the next available number of a dimension, box, or counter at TEX processing time and assign a symbolic name to it. `\newdimen`, `\newbox` and `\newcount` are the commands that allocate these numbers dynamically. It is safer to use dynamic allocation in a macro than to use a particular numbered box, counter, or dimension, since it prevents accidental reallocation.

Unfortunately, all of the commands in this useful set are `\outer`, which means that they cannot be declared *within* a macro. By making these dynamic allocation macros non-outer, we can then include them inside macros and only declare new counters or new boxes or new dimensions when they are needed.

Here is how to make these commands non-outer. Simply copy the original definition, supply a new control sequence name and define them without the `\outer` that originally preceded the definition. For example, the definition of `\newbox` was originally

```
\outer\def\newbox{%
  \alloc@4\box\chardef\insc@unt}
```

Here are the new versions, `\nonouternewbox`, etc.:

```
{\catcode`\@=11
\gdef\nonouternewbox{%
  \alloc@4\box\chardef\insc@unt}
\gdef\nonouternewdimen{%
  \alloc@1\dimen\dimendef\insc@unt}
\gdef\nonouternewcount{%
  \alloc@0\count \countdef \insc@unt}
\catcode`\@=12}
```

In the next section we will see these non-outer commands being used in a table macro, only making named boxes or dimension when needed. Macro writers may find other uses for this technique as well.

Fun with Csname

One of the really useful features of `\csname` is that control sequences can be expanded within the body of the `\csname...\endcsname` construction:

```
\expandafter
\def\csname\testmacro\endcsname{%
  ⟨definition⟩...}
```

Counters can be used:

```
\expandafter
\def\csname\testcounter\endcsname{%
  ⟨definition⟩...}
```

Counters with roman numerals can be used:

```
\expandafter
\def\csname\romannumeral\testcounter%
  \endcsname{⟨definition⟩...}
```

You can even make a definition out of numbers or other symbols that ordinarly are not allowed for a control sequence name:

```
\expandafter\def\csname 123&#\endcsname{%
  ⟨definition⟩...}
```

The only thing to remember here, is that any control sequence made with `\csname` that contains anything other than letters must be invoked as well as defined with `\csname`: `\csname 123&#\endcsname` is the way to use this macro.

Using `\csname` with `\expandafter` makes it possible to do all sorts of things that would not otherwise be possible. Some examples will be found in the following text.

Macros that define new macros using data supplied. One example involves having one macro define another macro where the name of the second macro depends on data supplied in the text.

In the example constructed below, `\usearg` takes the first two words as arguments `#1` and `#2`, reverses their order and uses them to make a control sequence name. This control sequence is then defined to be the complete name and address of the person whose name was used to form the control sequence name.

The order of the name is reversed so that the names of the new macros can be sent to an auxiliary file and be sorted alphabetically. The Appendix illustrates how a more elaborate form of this set of macros may be used to manipulate mailing lists.

`\obeylines` below changes the catcode of the end-of-line character (`^^M`) to 13 so it can be used as a argument delimiter in the definition of `\usearg`; `\obeylines` also defines `^^M` as `\par` so that every line ending seen on the screen is maintained when the text is printed:

```
{\obeylines
\def\usearg#1 #2^^M#3^^M^^M{%
  \expandafter\gdef\csname #2#1\endcsname
  {#1 #2\par #3}}
```

With this definition,

```
\usearg George Smith
21 Maple Street
Ogden, Utah 68709

\SmithGeorge
}
```

produces

George Smith
21 Maple Street
Ogden, Utah 68709

The Appendix contains a macro subsystem for processing and sorting address labels; it demonstrates this technique and many of the others discussed in this paper.

Macros that define new macros using a counter. Here is another use of `\csname`: In this case, it is used to define a macro that will itself define a new macro every time it is used, with the name of the new macro determined by a counter whose number is represented with roman numerals. This can be used to construct a series of macros that expand into areas of text to be reprocessed at the end of a document. For instance, this technique could be used to produce a set of slides from given portions of the text of a document.

In the following example, each time `\testthis` is called it will define a new control sequence. It makes the name of the new control sequence by advancing `\testcounter` which is operated on by `\romannumeral` to produce a new set of letters. These letters will appear in the name of the new macro.

Each control sequence is then sent to an auxiliary file, embedded in code to make the slide. (The slide formatting code is represented here as [[[ ]]]). At the end of the document the auxiliary file containing all the definitions can be input, to produce a set of slides.

```
\newcount\testcounter
\testcounter=501
    %% just to start with a large
    %% number to make into roman numerals
\newwrite\sendtoaux
\immediate\openout\sendtoaux
  \jobname.aux %% opening a file to write to
\def\testthis#1{%
  \global\advance\testcounter by1
  \expandafter\gdef\csname%
  \romannumeral\testcounter XYZ\endcsname{%
      [[[#1]]]}
\immediate\write\sendtoaux{%
\noexpand\csname\romannumeral\testcounter
XYZ\noexpand\endcsname}}
```

Here is an example of `\testthis` being used:

```
\testthis{This is the first bit of text...}
\testthis{This is the second...}
```

The code above writes the following lines into the `.aux` file:

```
\csname diiXYZ\endcsname
\csname diiiXYZ\endcsname
```

and when the `.aux` file is input, these commands produce

[[[This is the first bit of text...]]] [[[This is the second...]]]

Designing generic code with csname. Another really important use for `\csname` constructions is a way of making compact code for a section of a macro that repeats many times with a small variation each time. Table macros are often examples of this kind, since they tend to have repeating sections, one for each column.

We consider below some parts of a set of macros for table construction showing several ways that `\csname` can be used. The form adopted for the `\halign` command line, using a `&` immediately after the `\halign{`, will allow the specifications for this column to be used for each column in the table.

Using `\csname` with a counter allows this set of commands to be defined only once. Each new column entry will cause the `\colcount` counter to advance making the otherwise similar column definition use a new counter value inside the `\csname...\endcsname` constructions.

Non-outer dynamic allocation is used to name only those counters, dimensions, or boxes that are needed when the table is made up. A new set is declared for each column of the table. Since we don't need to guess ahead of time how many columns are going to be used, only those dimensions or boxes that are needed will be declared. In addition, `\ifdefined` (below) tests to see if the particular set of dimensions and boxes has been used in a previous table, and will only declare a new set if they have not been defined previously.

`\xtab` and `\dtab` involve counters only, so they can be used later in a `\csname...\endcsname` construction where it doesn't matter if the expansion will produce numbers as part of the control sequence. `\gtab` and `\vtab`, on the other hand, need to be used as ordinary control sequences which is the reason for the `\romannumeral` command that will produce letters instead of numbers when the `\gtab` and `\vtab` are expanded.

`\asizetab` and `\finishasizetab` will use these boxes and counters to actually set the table entries.

Here, finally, are some definitions for table construction:

```
\def\multipagetable{\global\firstcoltrue
  \halign\bgroup%
    &\global\advance\colcount by1\relax%
    \ifdefined{\the\colcount tab}{}{%
      \edef\xtab{\expandafter\csname
        \the\colcount tab\endcsname}%
      \edef\dtab{\expandafter\csname
        \the\colcount tabwide\endcsname}%
      \edef\gtab{\expandafter\csname
        \romannumeral\colcount
          gapped\endcsname}%
      \edef\vtab{\expandafter\csname
        \romannumeral\colcount
            vlinewd\endcsname}%
      \expandafter\nonouternewdimen\vtab%
      \expandafter\nonouternewbox\xtab%
      \expandafter\nonouternewdimen\dtab%
      \expandafter\nonouternewcount\gtab%
    }%
    \ifdefined{align\the\colcount tab}{}{%
      \edef\atab{\expandafter\csname
        align\the\colcount tab\endcsname}%
      \expandafter\nonouternewcount\atab}%
  \asizetab##\finishasizetab\cr}
```

A `\csname..\endcsname` construction defined using one counter can be invoked *using a different counter*, if that proves useful. Another part of the code for multipage tables uses a second counter to invoke macros defining boxes containing the column heads, used when the table continues over page breaks. Even though the original definition used `\colcount` as the counter to name the boxes, `\contcolcount`, another counter, can be used in another macro to invoke the same definition. When TEX expands `\csname...\endcsname` construction it produces a number as the replacement for the counter, so the name of the counter used doesn't affect the result. This might be helpful in cases where you don't want to change the value of one counter, but still wish to use a `\csname` construction that contains it.

Tips on Table Macros

`\everycr` is a TEX primitive for a token list. It functions similarly to `\everypar` or `\everymath` in that its definition will be used every time the named environment is present, in this case after every `\cr`. By setting `\everycr` equal to some definition we can insert a set of commands after every line in a table, since every line will end with a `\cr`. A simple example is this:

```
\everycr={\noalign{\hrule}}
```

which will insert a horizontal rule automatically after every `\cr` in the table. Once this possibility is discovered, the macro writer may realize that there are many other things that can be done with `\everycr`, such as including a set of conditionals that will call for horizontal lines with breaks in them, double horizontal lines, thicker horizontal lines, thicker lines under some columns but not under others, and so on.

You can include a counter which is advanced every time `\everycr` is called, and use that counter to determine how many lines have been used in the table, in order to stop and restart the table, making it possible to have a table that will continue

for hundreds of pages without TEX running out of memory.

Moreover, you can call for a small vertical skip in the \everycr definition which will allow the table to break over pages. If you use the following construction, your table will break over pages and a horizontal line will appear both at the bottom of the previous page and at the top of the new page, without the user having to know ahead of time where the table will break.

```
\everycr={\noalign{\hrule\vskip-1sp\hrule}}
```

When Doing Nothing is Helpful

The usual form of a macro with an argument is (in its most basic form) \def\example#1{#1}. There are cases in which **not using** the argument can be helpful when you want to get rid of something: \def\example#1{}.

You can use this principle to prevent large sections of text from being processed by TEX.

```
\long\def\ignorethis#1\endignore{}
```

Thus

```
\ignorethis
Here is some text that will be ignored...
\endignore
This is where \TeX\ starts printing text...
```

will produce

This is where TEX starts printing text...

You might want to use this macro in the process of debugging a document you are working on. All text between \ignorethis and \endignore will be ignored, making it possible for TEX to print only the part of the document in which you are interested. TEX will run out of memory after about 6 pages of text is picked up by the \ignorethis macro, depending on the implementation of TEX being used, but if you want to ignore more than 6 pages of text you can end the first \ignorethis with \endignore and enter a second \ignorethis ...\endignore.

A slight improvement, however, is needed to prevent TEX from complaining if an \outer command is found in the argument of \ignorethis. This is the error message which we would like to avoid:

```
! Forbidden control sequence found
while scanning use of \ignorethis.
```

We can avoid it by changing the catcode of the backslash to be that of a letter. Now there will be no commands processed until \ignorethis encounters \endignore and the catcode changes are turned off.

```
\long\gdef\ignorethis{\bgroup
  \catcode`\\=12 \catcode`\^=12 \finish}

{\catcode`\|=0 |catcode`|\=12
|long|gdef|finish#1\endignore{|egroup}%
}
```

Note that here, too, #1 is never used in the replacement part of the macro.

Getting rid of backslashes. Here is another example of an argument that is thrown away:

```
\def\stripbackslash#1#2*{\def\one{#2}}
```

which only uses the second argument, throwing away the first argument, in this case stripping away a backslash from a control sequence supplied by the user. \stripbackslash can then be used in another macro which needs a control sequence without its backslash to work correctly, for instance:

```
\def\newdef#1{\expandafter
  \stripbackslash\string#1* \one}
```

When this is used,

```
\newdef\testmacro
```

produces

```
testmacro
```

Instead of simply printing the control sequence without the backslash, \newdef can be rewritten to test to see if a given macro has already been defined. In this example, \newdef tests to see if the expansion of the control sequence \csname\one\endcsname, (where \one, was defined by \stripbackslash to be the control sequence supplied by the user minus its backslash) is equal to \relax. This takes advantage of the TEX convention that a previously undefined control sequence invoked in a \csname...\endcsname environment will be understood to be equal to \relax, whereas an already defined control sequence will not:

```
\def\newdef#1{%
  \expandafter\stripbackslash\string#1*
      %% \stripbackslash defines \one
  \expandafter
    \ifx\csname\one\endcsname\relax
      %% \one is expanded to be the
      %% control sequence the user supplied
      %% minus the backslash.
      %% If csname construction equals
      %% \relax, do nothing
    \else    %% Else, give error message:
      {\tt Sorry, \string#1 has already been
      defined. Please supply a new name.}
    \fi}
```

In the test below, notice that we do not get an error message for \cactus which hasn't been previously defined, but we do get a message for TEX, which is defined:

```
\newdef\TeX
\newdef\cactus
```

produces

```
Sorry, \TeX has already been defined.
Please supply a new name.
```

Not using boxes. Similar to not using arguments, there are times when setting a box and then **not** using it can be useful.

When writing a macro to make text to wrap around a given figure, we might want to use a test box to put a given amount of text in, say, a paragraph, which has been picked up as an argument to a macro. We can then measure the box to see if it will exceed the depth of the figure. If it does not, the box can be used as it is, but if it does, the box can be ignored and the argument re-used, with changed `\hangindent` and `\hangafter`, to allow the text to fit around the figure neatly. This works because text picked up as an argument to a macro does not yet have its glue set, so it can accomodate different line widths.

Another use for a box that is never printed is to use it as a container in which to expand a macro having symbols in the parameter field. For example, if the macro `\splittocentry` is defined by

```
\def\splittocentry#1-#2-#3{\gdef\one{#1}
  \gdef\two{#2}\gdef\three{#3}}
```

we can use it in a another macro to process an argument which may or may not include the hyphens, i.e.,

```
\setbox0=\hbox{\expandafter
  \splittocentry#2-{}-{}}
```

The hyphens that are necessary to complete the use of `\splittocentry` are supplied in the box but they will not print if the replacement for `#2` turns out to supply the hyphens already. Since `\one`, `\two`, and `\three` are globally defined (`\gdef`), their definition will be understood outside the box.

Some General Macro Writing Tips

There are several commands that can make the process of macro writing easier.

`\show` is a TEX primitive that will cause the definition of the macro it precedes to appear on your screen when you run TEX on a file that contains it. `\show\samplemacro` will cause the definition of `\samplemacro` to be appear on your screen, for example. `\show` can be temporarily included inside a macro to let you see what is being picked up as arguments. For instance, if

```
\def\test#1,#2{\def\one{#1}\def\two{#2}
\show\one\show\two....}
```

then

```
\test some, stuff
```

will help you see what is being picked up as argument `#1` and `#2`. In this example the results are obvious, but there are more complicated situations. For example, when one macro is looking at the contents of another macro, a test like this can quickly help you understand what TEX sees when it picks up an argument, a helpful debugging tool. It also has the advantage of giving you information at the time you TEX your file, saving you the steps of either previewing or printing the `.dvi` file.

`\show` will also send the definition of the macro that it precedes to the `.log` file, a feature which you can take advantage of when you are interested in redefining a Plain TEX macro. If you write `\show\raggedright`, for example, in a test file and run TEX on that file, the definition of `\raggedright` will appear in the `.log` file. You can then move those lines of code from your `.log` file to your macro file and you will have saved yourself the trouble of looking up the command in *The TEXbook* and copying it into your file. Now you are ready to make changes to the original macro.

A related command, `\showthe`, will give you the current value of a token list, like `\everypar`. Including `\showthe\everypar` in a test file can tell you what TEX sees as the current value of `\everypar` at that point in the file. You can also use `\showthe` to get the current value of a counter or dimension. You may want to include a `\showthe` temporarily in a macro you are developing, similarly to `\show`, as a debugging tool.

Finally, using `\message` in a conditional while working on a macro can give you helpful information. You could put this code in a headline, for instance, to be able to see the state of a particular conditional in the headline,

```
\headline={....%
  \iftitle
    \message{SEES TITLE, WIN}
  \else
    \message{NO TITLE, LOSE}
  \fi...}
```

or include a similar construction in the body of a macro while you are testing it. When you TEX the file you can quickly see if you are getting the results you were expecting.

Appendix

Code to Alphabetize an Address List

These macros demonstrate many of the techniques discussed in this paper. The macros process an existing an address list by taking the first line of each address, re-ordering the name with last name first, then turning the name into a control sequence which is sent to an auxiliary file. The user must alphabetically sort the auxiliary file. The resulting sorted file is then input back into the originating file and the whole address list will be transposed and printed in alphabetical order.

The user enters `\alphalist` at the beginning of an address list, and a blank line and `\endalphalist` at the end. `\alphalist` picks up the name, then makes a macro using the name (last name first) as the control sequence. This control sequence is sent to auxiliary file with the same name as the originating file and with an `.alf` extension. The file `filename.alf` must be sorted to produce `filename.srt`, using a sort routine on the user's system. If DOS, write

```
sort < filename.alf > filename.srt
```

`\endalphalist` checks to see if `filename.srt` exists, and if so, will `\input filename.srt`. The sorted list of control sequences will produce an alphabetized address list.

First we name dimensions and counters and set them to arbitrary sizes.

```
\newdimen\heightofentry \heightofentry=.75in
\newdimen\widthofentry \widthofentry=.3\hsize
\newcount\namenum
```

`\alphalist` makes every new paragraph start with the command `\look`. `\obeylines` will maintain the same line endings as seen on the screen.

```
\def\alphalist{\bgroup\obeylines
  \global\everypar={\look}}
```

First we discuss the definition of `\look`, then we will consider the macros used in its definition.

`\look` picks up the entire name. It then defines it as `\test`. `\test` is placed in `\box0` and expanded after `\throwawayjr` which defines `\fullname`. Then `\fullname` is expanded after `\takeapart` to define `\nameinrev`. `\nameinrev` is the name in reverse order; it is used as the name of a control sequence that defines the entire name and address. `\nameinrev` in a csname environment is also sent to an auxiliary file so that it can be sorted alphabetically. Here is the definition of `\look`:

```
{\obeylines
\gdef\look#1^^M#2^^M^^M{\def\test{#1}
  \setbox0=\hbox{%
    \expandafter\throwawayjr\test, {}}
  \global\namenum=0
  \expandafter\takeapart\fullname
  \obeylines
  \everypar={}
  \expandafter%
  \gdef\csname\nameinrev\endcsname{%
    \vtop to\heightofentry{\parindent=0pt
      \vfill\hsize=\widthofentry
      #1
      #2
      \vfill}}%
  \immediate\write\alphafile%
    {\noexpand\csname\nameinrev%
    \noexpand\endcsname}%
  \global\everypar={\look}}
}
```

Now we consider the commands used in the definition of `\look`.

To make the last name appear first in the command sent to the auxiliary file, we count the number of parts to the name ("Mr. R. G. Greenberg" has four parts, for example) and use `\ifcase` to select the correct order. After `\nameinrev` (for 'name in reverse order') is defined, it will then be used in the `\look` macro to create a control sequence by being expanded within a csname.

```
\def\makerightdef{\ifcase\namenum\or
  \or\gdef\nameinrev{\one}
  \or\gdef\nameinrev{\two\one}
  \or\gdef\nameinrev{\three\one\two}
  \or\gdef\nameinrev{\four\one\two\three}
  \or\gdef\nameinrev{\five\one\two\three%
    \four}
  \fi}
```

`\makedef` gives a control sequence name to the argument of `\takeapart` according to the number of times `\takeapart` is invoked:

```
\def\makedef#1{\ifcase\namenum
  \or\gdef\one{#1}
  \or\gdef\two{#1}
  \or\gdef\three{#1}
  \or\gdef\four{#1}
  \or\gdef\five{#1}
  \fi}
```

In order to make an `\ifx` comparison, we set

```
\def\aster{*}
```

`\takeapart` loops until it sees the `*`, which will be supplied in the `\throwawayjr` macro:

```
\def\takeapart#1 {%
  \global\advance\namenum by1
  \def\onex{#1}
  \makedef{#1}
  \ifx\onex\aster
    \makerightdef\let\go\relax
  \else
    \let\go\takeapart
  \fi\go}
```

We want to alphabetize according to the last name, and not mistakenly use 'Jr.' as the last name. The first argument ends when `\throwawayjr` sees a comma, which would normally occur before a Jr. or Sr. following a name. The second argument is never used, which is how Jr., or Sr., or III, are thrown away:

```
\def\throwawayjr#1, #2{%
  \gdef\fullname{#1 * }}
```

`\throwawayjr` is used inside a box that is never used, so we can supply the comma that ends argument `#1`, in case there is no comma in the name given. If a name is used that contains a comma, that comma delimits the first argument. Since the extra comma is in a box that is never invoked, the extra comma is never printed.

Here is code to open an auxiliary file whose name is the same as the file containing `\alphalist`, but with an `.alf` extension:

```
\newwrite\alphafile
\immediate\openout\alphafile=\jobname.alf
```

Now we have finished describing the commands needed to define the names and address and to send their macro names to the auxiliary file, and it is time to input the sorted list.

`\endalphalist` turns off the `\everypar` that was established with `\alphalist` and inputs the `.srt` file if it exists. Since all the definitions precede `\endalphalist`, when the `.srt` file is brought in with the csname control sequences in it, each control sequence will produce its defined name and address:

```
\def\endalphalist{\egroup
  \global\everypar={}
  \openin1 \jobname.srt
  \ifeof1 %
    \message{<<Please sort \jobname.alf
      to produce \jobname.srt >>}
  \else
    \immediate\closein1
    \input \jobname.srt
  \fi}
```

Example:

```
\alphalist
George Smith
21 Maple Street
Ogden, Utah 68709

Jacqueline Onassis
Upper East Side
NYC, NY

Mr. W. T. C. Schoenberg, Jr.
Travesty Lane
Culver City, Iowa

\endalphalist
```

This writes the following lines in the file `test.srt` after `test.alf` is sorted:

```
\csname OnassisJacqueline\endcsname
\csname SchoenbergMr.W.T.C.\endcsname
\csname SmithGeorge\endcsname
```

which will transpose the original list to print the names and addresses in alphabetic order.

The complete address list code.

```
\newcount\namenum
\newdimen\heightofentry \heightofentry=.75in
\newdimen\widthofentry \widthofentry=.3\hsize
\def\alphalist{\bgroup\obeylines
  \global\everypar={\look}}
{\obeylines
\gdef\look#1^^M#2^^M^^M{\def\test{#1}
\setbox0=\hbox{%
 \expandafter\throwawayjr\test, {}}
  \global\namenum=0
  \expandafter\takeapart\fullname
  \obeylines  \everypar={} \expandafter%
  \gdef\csname\nameinrev\endcsname{%
    \vtop to\heightofentry{\parindent=0pt
      \vfill\hsize=\widthofentry
      #1
      #2
      \vfill}}%
  \immediate\write\alphafile%
    {\noexpand\csname\nameinrev%
    \noexpand\endcsname}%
  \global\everypar={\look}}}
\def\makerightdef{\ifcase\namenum\or
  \or\gdef\nameinrev{\one}
  \or\gdef\nameinrev{\two\one}
  \or\gdef\nameinrev{\three\one\two}
  \or\gdef\nameinrev{\four\one\two\three}
  \or\gdef\nameinrev{\five\one\two\three%
    \four}\fi}
\def\makedef#1{\ifcase\namenum
  \or\gdef\one{#1}
  \or\gdef\two{#1}
  \or\gdef\three{#1}
  \or\gdef\four{#1}
  \or\gdef\five{#1}\fi}
\def\aster{*}
\def\takeapart#1 {%
  \global\advance\namenum by1
  \def\onex{#1} \makedef{#1}
  \ifx\onex\aster \makerightdef\let\go\relax
  \else \let\go\takeapart
  \fi\go}
\def\throwawayjr#1, #2{%
  \gdef\fullname{#1 * }}
\newwrite\alphafile
\immediate\openout\alphafile=\jobname.alf
\def\endalphalist{\egroup
  \global\everypar={}
  \openin1 \jobname.srt
  \ifeof1 \message{<<Please sort \jobname.alf
      to produce \jobname.srt >>}
  \else
    \immediate\closein1
    \input \jobname.srt\fi}
```

BIJLAGE Q

The Document Style Designer as a separate Entity

Victor Eijkhout & Andries Lenstra

Abstract

An argument for the need for a programmable meta format: a format that introduces a new syntactic level in TEX for document style designers.

TEX has a number of characteristics that set it apart from all other text processors. Its unsurpassed quality of text setting and its capabilities for handling mathematics are some of the more visible aspects. On a deeper level, however, the extreme programmability of TEX is just as big an asset. Any layout can be automated to an arbitrary extent. (It is strange that *The TEX book* gives almost no hint of this.)

The form such automation usually takes is what is called 'generalized markup'. The person who keys in the text has at his or her disposal commands that describe the logical structure of the document, and as little as is possible of the visual structure.

Document styles as they appear in LATEXare examples of this. Here the layout is not merely automated, it is completely hidden from the end user. In particular, the same input can produce widely different output by letting it be interpreted by different styles.

With LATEX, however, the problem is the production of document styles. This is a major task, and consequently most people use either the standard styles, or minor modifications of these. Furthermore, LATEX does not offer sufficient tools to produce even moderate variations on the layout of the standard styles.

For scientific use of TEX one can become reconciled to this situation. A scientist should be more concerned with the contents than with the looks of a document, so if there is a format that offers all the tools to get those contents across to the reader, the visual appearance is of secondary concern. We may conclude that, for scientists, LATEX fits the bill.

When a document is not a scientific article, however, the inflexibility of LATEX renders it useless. The alternative would seem to be plain TEX, but there the objection is the long and slow learning curve.

One way out of this dilemma is the 'front end to TEX' approach taken in *The Publisher* from Arbortext and *Grif* from Gipsi. Both systems present almost a 'wysiwyg' (what you see is what you get) interface to the user (the term 'wysipn', what you see is pretty neat, has been used), and allow altering style parameters via dialog boxes and menus. In both cases, however, programming the basic style structure and appearance is still less than trivial.

In this article we describe an approach which brings down the complexity of programming a style to that of using it. We propose a front end programming language for style design, which is itself implemented in TEX.

The 'Checklist' Approach to Style Definition

If one compares TEX to mouse-and-menu text processor packages, one runs into two basic characteristics of programming that constitute a disadvantage when learning TEX.

The obvious first point is that TEX has a *syntax*. Every TEX programmer knows that you can have no end of fun with missing or mismatched braces. Mouse riders are not bothered by this. One can, at most, click the wrong item, but one cannot click in the wrong way.

The second point is more subtle: programming involves and imposes algorithmic thinking. Consider the ordinary loop statement

```
for $i$:=$1$ to $n$
```

In many cases, all that is meant is

```
for all $i$ between $1$ and
        $n$ inclusive
```

Thus, the syntactic formulation contains a sequentiality that may semantically not be present. Similarly, the statements in a macro definition are sequentially ordered, even when the corresponding actions are in no such relation.

But even when actions are sequentially ordered, it should not always be necessary for a style designer to specify them in that same order. For instance, in the design of a list environment the amounts of white space above and below the list, and the amount of indentation of the list, should be specifiable in any order, even though they correspond to sequentially ordered actions. Also, the decision whether or not to break a page above a certain

type of heading should be specifiable at any point in the definition.

Such actions do not really correspond to design decisions, rather they are the specific incarnations of general parameters and switches. Thus, it would be a valid approach to style design to offer the person implementing the style a small number of basic constructs (these could for instance be headings, lists, and page layouts), each of which has a 'checklist' of parameters and switches controlling the final layout.

Abandoning sequentiality, and making most specifications optional with default values, is then a sensible way to facilitate TEX programming. An implementation of these demands could take the form of lists of keyword-value pairs. Such lists can be presented in extremely simple syntax, and as matching is performed by keyword instead of by position, such an approach would meet some of the criticism of algorithmic specification above.

As an example, the layout of unnumbered section headings could be given in Lisp 'property list' syntax as,

```
\defineheading:section
   (white_above 6pt plus 2pt minus 1pt)
   (white_below 6pt plus 2pt minus 1pt)
   (caption (size 12pt) (style bold)
            (shape ragged_right))
```

but any other syntax is just as valid. It is advisable, however, to steer clear of the idiosyncrasies of the pure TEX syntax.

Metaformats

Checklists may capture a large part of the variation in basic constructs, but for any set of parameters there will be a layout that eludes classification in terms of these parameters. Thus, it seems inevitable that a style implementer needs to do some programming. However, it is possible to make a very smooth transition between marking a checklist and programming macros. For this, we need the distinction between formats and 'metaformats'.

Let us denote by the term 'format' any collection of macros that gives the end user commands of a higher level than those of pure TEX. By 'metaformat,' we will mean a format such that the commands for the end user are in majority not defined in the format. Metaformats offer the tools with which a style implementer constructs the commands for the end user.

In a restricted sense, the LATEX format is a metaformat. A good example here is the `\@startsection` macro, which is basic to LATEX, and in terms of which commands like `\section` are defined in the style files. The parameters of this command are mainly numeric parameters determining the layout. One parameter functions as a switch.

As another example, the `\list` command, which is the basis for various environments in LATEX, offers the style designer the possibility for programmable extension. Such extensions are specified by passing a piece of TEX code as the second parameter, that is, LATEX does not really provide the style implementer with a separate syntactic level.

Calling LATEX a 'parameterized metaformat,' we can also envision 'programmable metaformats.' There is no objection against implementing a new programming language in TEXwhich would be easier to learn and to use.

The challenge is then to find primitives that allow formulation in a simple syntax and that are sufficiently powerful for producing a wide range of layouts.

By any other name . . .

One choice for the primitives of a metaformat is immediately clear: the boxes and glue of TEX itself. Any TEX programmer knows that you can do all typesetting with boxes and glue, so why not give them to a style designer? The problem of course is how to simplify their declaration. It is here that the writer of the metaformat takes certain decisions.

Consider, as an example, section headings such as they appear in the LATEX article style.

1. Section title

1.1. Subsection title

These take the form (but not the implementation) of two boxes on the same line: one containing the number, the other containing the text. Also the headings of the 'artikel' style can be described thus.

1. Section title

1.1. Subsection title

But the box of the number then has a prescribed width. In both cases, however, the sum of the width is the text width. We can therefore imagine that these two layouts were specified as

```
\defineheading:section
    (inline
       ((value sectionnumber)
        (spaces 2))
       (title))
```

for the standard LATEX heading, and

```
\defineheading:section
    (inline
       ((value sectionnumber)
        (FillUpTo labelwidth))
       (title))
```

Here the metacommand 'inline' is just an `\hbox to \hsize` in disguise. It takes all boxes and sets them at natural or specified width, and the width of the box containing the actual heading is calculated to fill the remainder of the text width.

Headlines and footlines provide a nice example of how a programmable metaformat can make intelligent use of

TEX's glue. Like section headings, footlines are a disguised `\hbox to \hsize`. A footline with just a left aligned page number can be specified like

```
(footline (value pagenumber))
```

where the format supplies a trailing `\hfil` to prevent an underfull box. Cases where the number should be right aligned can be specified as

```
(footline (whitespace fillerup)
          (value pagenumber))
```

where the 'whitespace' is an `\hfill`, squashing the trailing `\hfil` at the end of the line.

A syntax and instruction set, such as sketched in these examples, tax the programming capabilities of the format designer, but not those of the style designer. In effect, the format designer implements a new programming language on top of TEX with a simple syntax, a small instruction set, and at the same time, sufficient generality to produce a wide range of layouts.

Obviously, a smaller instruction set makes learning the format easier. Another advantage is less immediately clear: it reduces the chance of errors. More compact instructions will likely have a more defined function; so on the one hand the style implementer need not specify those actions that must be taken anyway, and on the other hand the format can perform some consistency checking on the intentions of its user.

Conclusion

We have argued the need for a 'programmable metaformat:' a format that introduces a new syntactic level in TEX for document style designers. Such a level should probably have a different syntax from pure TEX, and it should contain a relatively small instruction set in which the actual user macros are written. We have indicated that elements of such a format can, to a large extent, be captured in 'checklists.' Others can take the form of intelligently disguised TEX primitives. We believe that a format incorporating these principles is possible, and that it can be taught to people with little knowledge of TEX.

As an illustration of these ideas we present the style definition of this article in the 'Lollipop' format.

Example

The following piece of code is the style definition for this article given in the 'Lollipop' format.

```
\typeface:Computer
\fontsize:10 \fontstyle:roman

%declare \parindent
\distance:indentation=20pt
%white space around text elements
\distance:whiteline=6pt plus 2pt
         minus 2pt
\distance:leadingwhite=whiteline
\distance:trailingwite=0pt

%lots of defaults used here
\defineheading:section
         fontstyle:bold stop

%very simple page layout
\definelayout:twocolumn
    height:22.5cm width:16.5cm
    band:start column
         whitespace:0.6cm
         column
         band:end
    stop
\twocolumn %install output routine

%paragraph shape
\defineparagraph:flushright indent:yes
    rightjustified:yes stop
\flushright %install paragraph shape

%bibliography
\definelist:literature counter:1
    item:left
         litteral:[ value:itemnumber
         litteral:]
    item:end stop
\externalreferences:yes
```

Bibliography

Braams, Johannes, Victor Eijkhout, and Nico Poppelier, "The development of national LATEX styles," *TUGboat*, Vol 10(4), 1989.

Grif manual / Les languages de Gipsi, Gipsi 1989

Knuth, Donald E. *The TEX book*, Addison-Wesley Publishing Company, 1984.

Lamport, Leslie. *LATEX, a document preparation system*, Addison-Wesley Publishing Company, 1986.

BIJLAGE R

The Dutch national LaTeX effort

Johannes Braams, Victor Eijkhout & Nico Poppelier

Abstract

In this article an overview is given of the activities of working group 13 (WG13) of the "Nederlandstalige TeX Gebruikersgroep" (Dutch TeX Users Group). This working group is also called "Neerlandica", and is interested in anything that has something to do with using LaTeX (and TeX) in a non-American environment. The topics tackled so far range from the design of a page layout suitable for A4 paper by adapting the American layout of `article.sty` to Dutch typographical tastes, to the implementation of a new letter style called "brief".

Introduction

At its founding meeting the NTG decided to establish a number of "working groups", which would tackle some of the problems encountered by members of the NTG. Some of the subject are "education", "drivers", "TeX for personal computers". At the second NTG meeting another working group was started. The task of this group was to investigate the TeXnical problems Dutch-speaking LaTeX users encountered and to suggest solutions to these problems.

The first activity of this group was to decide on which problems would be tackled and in what order. We had a number of subjects:

- The original LaTeX document styles are designed for American-sized paper. The dimensions of this paper differ from A4 paper, which is used most commonly in Europe. The available style option files at the time were not very satisfactory.
- In Dutch texts, commands that produce text, like `\chapter` or `\abstract`, should produce *Dutch* texts instead of *English*. A way to solve this problem had already been pointed out by Hubert Partl *et al* [5].
- The design of the standard document styles provided by Leslie Lamport is very "American" and, at least to our Dutch eyes, a bit "loud". We decided to develop replacement styles that would be more adapted to Dutch typographical standards.

After identifying these tasks we set ourselves a short-term and a long-term goal. The short-term goal was to provide the members of the NTG with an acceptible page-layout, adapted to A4 paper and to provide a document-style option that redefines commands like `\chapter` to use macros like `\chaptername`. This seemed a reasonably simple task that would be welcomed by a lot of Dutch LaTeX users. The long-term goal was to develop new document styles, plug-compatible with the styles provided by Leslie Lamport. To this we added the need for the implementation of a document style for letters that follows Dutch guidelines for the design of letters.

In the following sections we will discuss these topics in more detail. Some of this work has led to articles published or to be published in *TUGboat*.

Page layout for A4 paper

As described before one of the first problems tackled was that of adapting the standard document styles to A4 paper. The solution we sought was to provide a document style option file, and not to modify document styles for this. This was done mainly to avoid having to maintain extra document styles that differ only marginally from the originals.

Obviously, we were not the first LaTeX users to identify the problem and we knew that document-style options were available in the international TeX community. However, as we were not satisfied with the results these options gave us, we decided to adapt them. This has led to the file `A4.sty` described in [9]. This file started as a combination of two options that were already available. The combination of these two files didn't satisfy our demands, which were:

1. The width of the text should be chosen in such a way that no more than sixty to seventy characters appear on a line of text;
2. When a document is printed two-sided, the texts printed on both sides of one sheet of paper should overlap;
3. The "inner" margin of the document should be wide enough to allow for the binding of the document;
4. The "outer" margin should be wide enough for marginal notes.

After the publication of our article on `A4.sty`, we received a comment that it doesn't handle texts with more than one column correctly. We still have to look into

this, but it is a nice example of Gödels principle [11][1].

The "loud" American design of `article.sty`

As discussed before, the modification of the design of the standard document styles was a long-term project. As a first attempt a document style option "`sober.sty`" was developed. This file modifies the amount of white space around section heads and lists. It also modifies the fonts used for the various levels of section heads.

A more fundamental approach was the development of new document styles, derived from `article.sty` and `report.sty` [7]. The design of these "new" styles was based on discussions with a Dutch typographical designer and some books about typographical design [3, 4]. One of the ideas behind the redesign of `article.sty` was to minimize the number of "implied left margins". By an implied left margin we mean a non-zero distance from the actual left margin that is used in more than one element of the document. Examples of implied left margins are:

- the paragraph indentation,
- the left margins of items in an "itemize" or "enumerate" list construct,
- the left (or right) sides of the numbers and labels in such list constructs and
- the left side of the text of a section heading.

In the standard styles of LaTeX all of these four distances are independent and are different from one another. In the style we have developed it was decided to strengthen the visual coherence of the layout by taking the same value for each of them whenever possible.

Another idea that was implemented with `artikel1.sty` is that the white space separating a section heading and the text following it should bear some simple relation to the baselineskip, and should not have any stretch.

A third major modification in the design of the article style is a new layout of the table of contents. Both Treebus [3] and Miles [4] are very explicit in their opinion about the layout of a table of contents. They find a layout like the one implemented in `article.sty` "old fashioned" and even confusing. So, a completely different layout was implemented.

As the new layout of the table of contents met with some reservations from users, the old layout is still available through an option.

Besides `artikel1.sty` we also implemented two other article styles that have a layout that differs from the layout of `artikel1.sty`. In `artikel2.sty` we implemented a layout that was designed to show what can be accomplished by modifying a few parameters in a document style. The third article style is like `artikel1.sty`, but with one major design decision changed: in `artikel3.sty` paragraphs are not indented, but they are separated by vertical space.

The design of the document styles `artikel1` and `artikel3` is also implemented for reports, as `rapport1.sty` and `rapport3.sty` respectively, and there is a `boek` style based on the design of `artikel1`.

Modifying the standard styles without modifying them

As discussed in the introduction, one of the topics WG13 should be working on was a solution to the problem of English terms appearing in Dutch texts. The basic idea behind our first solution was already implemented in `german.sty` [5]. We adopted the idea and created `dutch.sty`. In some ways this file is much simpler than `german.sty`, in another way it is more complex. A major difference between the two files is that `german.sty` just provides parameters and parameter values for the various terms and states in the comment that the user should provide modified document style files that make use of these parameters. The file `dutch.sty` also contains the necessary redefinitions for the various LaTeX macros. This implies that `dutch.sty` can be used with the standard *unmodified* document style files as they are included in every LaTeX distribution.

While implementing `dutch.sty` it occurred to us that it had some code in common with `german.sty`. This, combined with several discussions at the 1989 EuroTeX meeting in Karlsruhe led to the idea of building a more universal system of style option files, called the babel-system [10].

The babel system consists of one file, with macro definitions common for all languages and a language-specific file for every language that is to be part of the system. It offers the possibility to switch between languages while processing a multilingual document. Because we wanted this system of style option files to be compatible with the original `german.sty` the files are implemented in such a way that they can also be used with plain TeX. This is useful because the language-specific files can (and do) contain more than just parameters for LaTeX terms. For instance, for the German language as well as for the Dutch language we have an extra active character. This active character is, among other things, used for controlling the hyphenation of words containing accented letters or explicit hyphens by inserting `\discretionary` commands.

All the user needs to specify is the (main) language used in his document as an option to the `\documentstyle`

[1] See the chapter about the *Contracrostipunctus*.

command. This will instruct TeX to read the appropriate language-specific file. This file checks whether the core of the babel system, `babel.sty`, has been read before. If this turns out not to have happened it `\input`s `babel.sty`.

When the user wants to add the definitions for another language to the environment in which his document is processed he can use the command `\addlanguage`[2] with the name of another language-specific file as an argument. So the preamble of his document might look like:

```
\documentstyle[11pt,dutch]{artikel1}

\addlanguage{germanB}

\begin{document}
```

So, this example document contains Dutch and German texts. Because the function of the extra active character (") is different for `dutch.sty` and `german.sty` the user wants to switch this definition when he starts a German part of the document. This can be achieved by adding

```
\selectlanguage{germanB}
```

in front of the German text. When he wants the Dutch settings to be restored he simply uses the same command with the parameter `dutch`.

As the babel system has been developed in a pre-TeX 3.0 environment, it doesn't use any of the features of TeX 3.0. Perhaps the switching of hyphenation tables and maybe other TeX 3.0 features might be added to the definition of `\selectlanguage`.

The design of a new letter style

In one of its meetings the "Neerlandica" group decided to try to implement a document style for letters that should conform to Dutch standards for the layout of letters. We have consulted four such standards, from the Dutch standardization institute, the Nederlands Normalisatie Instituut (NNI). They were:

- NEN-1026 for letters,
- NEN-1025 for envelopes,
- NEN-3162 for the structure of documents and
- NEN-3516 for the design of forms.

The result is a design that can not be judged by taste: it just implements the standards. This seems quite rigid, but some freedom is left to the user to adapt certain parts of the layout to his own wishes.

This new style is *not* "plug compatible" with the LaTeX `letter` style, although equivalents of some of the macros from `letter.sty` have been provided. The main reason for its "incompatibility" with `letter.sty` is that we have quite a lot of new user commands to either modify some parts of the layout or to fill in some of the fields in the "reference lines". The reference lines contain fields such as "Your letter of" and "Date".

Some of the features of this document style are:

- If the user doesn't have printed letter paper he can provide his own letterhead by writing his own macro `\briefhoofd` or he can use the macro `\maakbriefhoofd` to adapt the default letterhead provided with the document style.
- An option is provided to print short horizontal rules on the sides of the paper as an indication where to fold the letter.
- The address is positioned such that it can be used with "window envelopes". The window can be either on the left side (default) or on the right side of the envelope.
- Provides a user-command `\voetitem` for providing information about the sender at the bottom of the letter.

An example of this document style can be found at the end of this article.

Conclusion

Working group 13 of the NTG has had a busy year. We have

- produced a number of replacement document styles for the standard LaTeX document styles,
- developed a new document style for letters that implements Dutch standards for the layout of letters,
- presented a new and improved document style option file for use with A4-size paper and
- produced a document style option file system that supports multiple languages in one document and provides additional features for specific languages.

In short: the category code of this working group has been `\active`.

References

[1] Donald E. Knuth, *The TeXbook*, Addison-Wesley, 1986.

[2] Leslie Lamport, *LaTeX, A document preparation System*, Addison-Wesley, 1986.

[3] K.F. Treebus. *Tekstwijzer, een gids voor het grafisch verwerken van tekst*, SDU Uitgeverij ('s-Gravenhage, 1988). A Dutch book on layout design and typography.

[4] John Miles, *Ontwerpen voor Desktop Publishing, Alles over layout en typografie op de personal computer*, Uitgeverij Mingus, Baarn 1988, Dutch translation of *Design for Desktop Publishing: a guide to layout and typography on the personal computer*, Fraser, London 1987.

[2] `\newlanguage` would be more appropriate, but this has become a TeX primitive, so the macro was renamed.

[5] Hubert Partl, *German TeX*, *TUGboat* 9 (1988) #1, p. 70–72.

[6] Leslie Lamport, in: TeXhax Digest, Volume 89, #13, 17 februari 1989.

[7] Johannes Braams, Victor Eijkhout and Nico Poppelier, *The development of national LaTeX styles*, *TUGboat* 10 (1989) #3, p. 401–406.

[8] Joachim Schrod, *International LaTeX Is Ready To Use*, *TUGboat* 11 (1990) #1, p. 87–90.

[9] Nico Poppelier and Johannes Braams, *A style option to adapt the standard LaTeX document styles to A4 paper*, *TUGboat* 11 (1990) #1, p. 98–103.

[10] Johannes Braams, *Babel, a multilingual style-option system for use with LaTeX's standard document styles, To be published in TUGboat*.

[11] Douglas R. Hofstadter, *Gödel, Escher, Bach: een eeuwige gouden band*, Uitgeverij Contact, Amsterdam 1985, Dutch translation of *Gödel, Escher, Bach: an eternal golden braid*, Basic Books, New York 1979.

NTG

Nederlandstalige
TEX
Gebruikersgroep

NTG • Postbus 394 • 1740 AJ Schagen

Jan TEXer
Overfullplein 10000
Baselinestad

Uw brief van	Uw kenmerk	Ons kenmerk	Datum
		NTG-JB 90/001	21 april 1994

Onderwerp: Contributie NTG over 1990

Geacht lid,

Langs deze weg wil ik U verzoeken Uw contributie voor het lidmaatschap van de NTG aan ons over te maken op girorekening 1306238, ten name van de Nederlandstalige TEX Gebruikersgroep te Waddinxveen onder vermelding van 'lidmaatschap 1990'. Wij stellen het op prijs indien U dit vóór 1 maart aanstaande doet. In dat geval bent U ervan verzekerd dat U het verslag en de bijlagen van de laatste NTG-bijeenkomst in Tilburg zult ontvangen.

Wellicht ten overvloede, de contributie voor een persoonlijk lidmaatschap bedraagt f 75,–, de contributie voor een instituutslidmaatschap bedraagt f 200,–. Een instituutslidmaatschap geeft het recht om drie personen aan te wijzen die alle informatie die aan de leden wordt verstuurd ontvangen. Van die drie personen dient één persoon te worden aangewezen als rechtsgeldig vertegenwoordiger van Uw instituut, een ander als vervangend vertegenwoordiger.

Indien Uw instituut een instituutslidmaatschap neemt verzoek ik U ons de bovengenoemde informatie te doen toe komen. Dit kan schriftelijk zowel als elektronisch. U kunt ons elektronisch bereiken via één van de volgende e-mail adressen: `JL_Braams@pttrnl.nl` ofwel `VANNES@ecn.nl`.

Hopende U van voldoende informatie te hebben voorzien verblijf ik,

Hoogachtend,

Johannes Braams
penningmeester NTG

BIJLAGE S

TUGboat production: TEX, LATEX, and paste-up

Barbara Beeton

American Mathematical Society
Editor, TUGboat

Introduction

TUGboat has now completed more than ten years of publication. Starting with TEX78 and an electrostatic printer and progressing through increasingly versatile software and hardware, the authors have kept us challenged, both with the content that the reader sees and the little tricks that happen "under the covers". This talk will be a survey of some of the milestones of *TUGboat* production, our editorial philosophy, what we've learned about what TEX can and cannot do, and some advice to authors and production editors of other publications.

History

TUG and *TUGboat* came into being together. An inaugural meeting of the TEX Users Group was held at Stanford University in February 1980; it was attended by about 50 people. The reaction to this new tool—TEX—was immediate and enthusiastic. One of the first projects proposed for TUG was the publication of an occasional newsletter, and it was immediately decided to call it *TUGboat*.

Bob Welland, a mathematician at Northwestern University, volunteered to handle the editorial responsibilities. Production would be done at the American Mathematical Society (which, together with the Stanford TEX Project, had arranged the first TUG meeting); this task was given to me, as I had more experience with TEX than anyone else at the AMS, although I had never been responsible for either editing or journal production.

The AMS before TEX / TEX before the AMS

A little "prehistory" might be useful here. The AMS had been using several proprietary photocomposition systems to prepare an increasing proportion of its research journals since about 1975, and for such "administrative publications" as its membership list and publications catalog for even longer. (The first typeset membership list appeared in 1971.)

Beginning in the early 1960s, the AMS had begun to investigate the possibilities for computerized composition of mathematical text. Most early computer-based typesetting systems were designed for newspaper production; many of them had quite strong pagination capabilities, but minimal ability to handle anything but ordinary text. Several systems appeared in the early 1970s that were able to deal with mathematical notation as well as text; however, they either required dedicated computer hardware or were unable to handle pagination, or both. In addition, all the typesetting software examined was proprietary, usually expensive (*troff* cost almost nothing for a university, but nonacademic users had to pay quite a high fee), and frequently had an obscure user interface.

The AMS was also looking at methods for incorporating large volumes of bibliographic material into a forerunner of today's database collections. For such a use, it was clear that logical, as opposed to typographic, tagging was the only sensible method for organizing these data. Even here the software must deal with mathematical notation—titles and abstracts of mathematical literature contain mathematical expressions, and to paraphrase or eliminate this notation could render much of the data meaningless.

In an effort to be part of the solution, representatives of the AMS (including me) participated for several years during the early 1970s in the GenCode project of the Graphic Computer Communications Association. GenCode was the forerunner of what later emerged as the Standard Generalized Markup Language—SGML.

At the annual meeting of the AMS in January 1979, the invited Gibbs Lecturer was Donald Knuth. The topic he chose to speak on was mathematics and computer science in the service of mathematical typography, and the new system he called TEX. The chairman of the AMS Board of Trustees, Richard Palais, was immediately attracted to the possibilities of the system that Knuth described. In addition to its ability to typeset mathematics and produce fairly sophisticated page layouts, this system was public, so it could be used by ordinary mathematicians who had access to computers at their universities.

The wheels were set in motion for AMS to become one of the first nonacademic users of TEX. This decision re-

sulted in (among other things)

- acquisition of a new computer;
- support of a group to spend the month of July 1979 at Stanford for the purposes of
  - learning TEX,
  - bringing TEX back to AMS and installing it at AMS headquarters,
  - implementing TEX macros to produce two particular AMS publications: the Society's membership list, and the issue indexes of *Mathematical Reviews*;
- the development of $\mathcal{AMS}$-TEX, a macro package intended to simplify input of complex mathematical expressions;
- the commitment to use TEX for all AMS publications prepared in-house.

A very short history of TUG vis à vis the AMS

Within a year after TEX came to the attention of the AMS, it was realized that widespread support of this public-domain software would best be gained through a strong community effort. This in turn led to the formation of TUG. TUG was initially funded by the AMS and supported by AMS personnel, but after several years it had grown sufficiently strong to acquire its own personnel and incorporate as an independent society.

There has not been any formal connection between TUG and AMS since 1984. TUG still purchases some services from the AMS, such as the typesetting of *TUGboat*, and AMS personnel continue to participate in TUG meetings and as members of TUG's Board of Directors. However, the two organizations are now entirely separate.

The first issue of *TUGboat*

Articles for the first issue were written mostly by people associated with the AMS, members of the Stanford TEX Project team, and their associates.

Some very primitive macros, based on `basic.tex` (the prototype of `plain.tex`), were developed to handle the formatting of a few obvious textual elements—title, section, etc. These macros were made into a style file that was used to prepare copy for most of the authors with AMS connections. A look at one of the article files created for the first issue shows a philosophy that hasn't really changed even today, namely the use of tags that identify objects, not coding to specify what they should look like.

```
\title Publishing \&\ \TEX\cr
\\Ellen E. Swanson\cr\end
 ...
\parhead The Manuscript.\end
 ...
\parsub Copy editing.\end
 ...
```

Explicit typographic coding was used in only three places in this article to obtain a desired format: in one itemized list, in the bibliography heading, and to obtain the desired vertical skip before the signature block at the end. This is typical of the articles prepared or processed at AMS.

Mike Spivak submitted an article describing his new macro package, $\mathcal{AMS}$-TEX, which he was creating at the request of the AMS. $\mathcal{AMS}$-TEX has its own "preprint" style, and Spivak's article was formatted with an adaptation of that style. Like the prototype macros for *TUGboat*, the preprint style uses mostly logical tagging, but the article reverted to raw TEX code for explication of $\mathcal{AMS}$-TEX's input notation.

Processing at the AMS was done with TEX78, which was essentially Knuth's own version of the program; this was written in the SAIL programming language (which runs only on DECsystems 10 and 20), and converted by David Fuchs from the WAITS operating system on Stanford's DEC-10 to run on a DEC-20 under the TOPS-20 operating system. Output was prepared at 130% on a Varian 9211 (a 200dpi electrostatic printer that used liquid toner); this was reduced photographically when making the printing plates to improve the quality.

One figure, illustrating the position of the letter "q" in its font box, could not be generated with TEX (even now, the largest Computer Modern font available at the AMS isn't large enough), and was drawn onto the TEX-generated box in India ink from an enlarged photocopy. The cover art and the name "TUGboat" on the title page were likewise pasted up from non-TEX copy.

A letter to the TUG Chairman, Richard Palais, was reproduced from typewritten copy as received. (Nowadays, if we receive a letter only on paper, we convert it to a file and process it as if it arrived electronically.)

A long article describing an indexing facility for TEX was supplied as camera copy by the authors and included as an appendix to the first issue. The camera copy was prepared on a Versatec printer (another 200dpi electrostatic printer), and uses what appear to be Times Roman fonts rather than Computer Modern. I find this of particular historical interest because it shows that the use of non-METAFONT fonts with TEX is as old as TEX itself.

The *TUGboat* editor, Bob Welland, worked from paper copy prepared by the authors, or by the production staff at AMS (me). Nearly all communications were by paper mail or phone. Some files were provided on magnetic tape (diskettes did not become a reasonable alternative until much later). I was fortunate to have been "adopted" by the Stanford TEX Project to the extent of having guest accounts on the SAIL and Score computers, and thus some limited communication was possible with the Stanford TEX community and other authors who had Arpanet accounts. (I gained access to these guest accounts by long distance phone calls and Kermit. This was much more complicated than using the AMS Internet connection that is now available. Even so, it was definitely the

method of choice for obtaining electronic files when that was possible for authors.)

Once all the editorial changes had been made and camera copy assembled, including pages for a separate mailing list, the copy was delivered to the printer and duly appeared in print in October 1980, eight months after the first meeting.

Progress

The next few issues grew in size as TUG itself grew, and the affiliations of authors and the equipment available to them became more varied. *TUGboat* 2, no. (1) contained a self-referential article:

```
\title HOW TO PREPARE A FILE\cr
FOR PUBLICATION IN TUGboat\cr
\\Barbara Beeton\cr
American Mathematical Society\cr\end
```

This showed the tagging scheme used for the (somewhat improved) *TUGboat* macros, and encouraged authors to create their articles "on a computer file and submit [them] on magnetic tape." (We didn't yet offer to provide the macros to authors for their own use.)

Before *TUGboat* 2, no. (1) was sent to the printer, a new typesetter, an Alphatype CRS (with the phenomenal resolution of 5333dpi) had been installed at the AMS. It was not yet ready for extensive production, but one sample page was included in the issue, along with sample output of the same page from a variety of other (dot matrix, electrostatic, and laser) printers.

All the regular articles in *TUGboat* 2, no. (2) prepared at the AMS were output on the Alphatype. This included one article by Brendan McKay that demonstrated how very small dots (actually tiny `\vrule`s) could be used in conjunction with ordinary rules and symbols from various fonts to create "pictures". The strain on memory capacity was severe (as it is now with PICTEX), but watching the output emerge from the film processor was well worth the effort.

By the end of 1982 I had become quite dissatisfied by the frequent necessity of either shrinking the type size of the contents to fit on the back cover, or continuing the listing on an inside page. *TUGboat* has always been subdivided into major sections, providing a logical structure for the contents, and in the early issues, the section headings appeared in the tables of contents above the listings of articles in those sections. Breaking the contents into two columns, with section headings to the left (right aligned) and article listings to the right (left aligned) eliminated enough extra lines to reduce the list to a single page. Unlike a book or report, *TUGboat* doesn't require numbered chapters or sections, so moving the page numbers to the left of the article titles didn't cause any visual conflicts; it also got rid of the dot leaders, which I wasn't particularly fond of either. The "new" contents style first appeared for *TUGboat* 4, no. (1); the final adjustment, moving the title and issue information from the top center so that it is left aligned with the article title listing, was made for *TUGboat* 5, no. (2).

Changing of the guard

By the end of 1982, Bob Welland found that he was unable to continue as editor, and effective with *TUGboat* 4, no. (2), the title as well as the editorial responsibilities became mine. *TUGboat* 4, no. (2) incorporated two innovations in honor of this event. The first (which has not been repeated) was the accidental omission of the name *TUGboat* from the title page of the issue. The second was to number pages consecutively within a volume rather than starting over with each issue; not only did this simplify references to items from *TUGboat*, but it also eliminated a source of confusion when back issues are reprinted in full volumes.

The old SAIL implementation of TEX78 continued to be upgraded periodically at AMS to fix bugs. Since that was working well in production, we never attempted to install TEX80, the first Pascal implementation, and in fact, TEX78 continued to be used for AMS production until late 1986. This is not as backward as it may sound, for two related reasons. The first is that TEX78 was by then a stable system, while TEX80 was a prototype, with frequently changing syntax, and was almost certain to be replaced, as indeed it was.

The second reason for not upgrading more quickly was built-in delays in AMS journal publication procedures. AMS policy requires that authors be given a chance to review their papers after typesetting, and sometimes authors take many months to return the copy for publication; two years used to be the cutoff, after which a paper would be assigned to a journal issue for publication without approval. A typesetting program with unstable syntax is not suitable for use in such an environment.

TEX82 owes its name to the date when Knuth began the reimplementation of TEX in `WEB`, a process that was brought to fruition with the publication of the five-volume *Computers & Typesetting* series in May 1986.

The first use of TEX82 in *TUGboat* was in volume 5, no. 1, in an article by Knuth on TEX incunabula.[1] The proportion of TEX78 to TEX82 items dwindled for several issues, and the last article that required TEX78 appeared in *TUGboat* 6, no. (2) (July 1985, still nearly a year before the debut of *C&T*).

The gradual approach was taken to permit time for careful reimplementation of the *TUGboat* macros, not simply a translation from old to new syntax. The two are

[1] *Incunabula*, from the Latin for "in the cradle", designates books printed before a.d. 1501; hence works from the early period of a technology.

quite different, after all—the final list of differences between TEX78 and TEX82 takes 26 *TUGboat*-sized pages plus a three-page, three-column index.

A few more milestones

At the same time, the entire appearance of *TUGboat* was re-examined, and suggestions solicited from anyone who turned up who seemed to have some design competence. Richard Southall's pre-TUG meeting short course in August 1984 was one source of ideas. (*TUGboat* had been offered, quite seriously, as a object for criticism, when that course was first being arranged.) The style of Southall's published lecture notes in *TUGboat* 5, no. (2), which is quite different from everything else in *TUGboat*, was intended to demonstrate the principles that he had expounded. This endeavor was relatively successful, as judged by Southall, failing principally in having centered text rather than asymmetrical margins. The particular style used for this article was not adopted in full, but the ways of looking at how conceptual structures are embodied in graphical form have been reconsidered every time a format change has been contemplated. I recommend this article and the references it cites for anyone who wishes to learn more about the traditions of typography, and why traditionally made books look the way they do.

One particularly interesting item from a quite different point of view is the set of benchmarks from a review of technical word processors made by members of the Boston Computer Society's PC Technical Group; this appeared in *TUGboat* 6, no. (3). The full review, which appeared in the *Notices of the AMS*, examined fourteen products (including two implementations of TEX) that could be used on IBM PCs and compatibles to prepare technical material for distribution. Of the fourteen, only four (including both TEX implementations) were able to produce the nine distinct benchmark samples, and only two of those (one TEX implementation failed) were able to combine all the samples and run them at once without a failure of either pagination or memory space. Furthermore, TEX output was judged to be in a class by itself, not surprising for a system designed to produce output of typographic quality. On the other hand, nobody said that using TEX would necessarily be easy, and the benchmarks demonstrate that. The file that produced this article is still one of the most stressful that has ever been processed for *TUGboat*, and I have recently provided a copy to be used in benchmarking the current crop of PC implementations of TEX; the results will be appearing in an upcoming issue.

The guest-edited issue

TUGboat 7, no. (1) was a first in two ways: It was designed by a professional designer, and it was the first issue to use the `cm` versions of the Computer Modern fonts produced by METAFONT84. The time spent on this issue by the guest editors (David Kellerman and Barry Smith) and by me was more than double that required for any previous issue. Part of this is undoubtedly due to the development of an entirely new set of macros, and part on the fact that creation of the `cm` fonts for the Alphatype required double METAFONT processing—only METAFONT79 could generate fonts in the internal format required by the Alphatype, so 5333dpi bitmaps output by METAFONT84 had to be reprocessed for input to METAFONT79, etc., etc. The fact remains that the new macros, though more elegant and capable of a wider variety of effects than the old ones, required more time and attention in production. Since the position of *TUGboat* editor is a volunteer post, anything that increases the time required to produce an issue is counterproductive.

A couple of other important problems surfaced during the production of this issue. The first is the inability of TEX to know the attachment point of a footnote when a multi-column page has been divided by `\vsplit`. The second is more philosophical: What does one do when the style decrees that each article start on a new page, and an article ends with just a few lines on a page by themselves?

One other design question arose with the issue: What is appropriate content for the back cover? My own preference is to put the table of contents there, so that I can quickly find an item that I'm looking for. This serves such a useful function that there's no chance that an advertiser is going to get far asking for his ad to go on the back cover, even for a substantially higher price. But for *TUGboat* 7, no. (1), the designer wanted to keep the back cover blank except for the issue identification. This is a dilemma to which there is no objective solution, based as it is so clearly on personal preference.

In the absence of solutions to these problems, the next issue returned to the old style, but with a few small changes (such as in the format of the device driver charts) based on some easily adapted features of the new macros.

And still more milestones

In spite of the fact that LATEX is the premier example of logically structured input for TEX, the first use of LATEX in *TUGboat* production didn't happen until *TUGboat* 8, no. (3), with the article on halftones by Adrian Clark. Before this, any articles submitted in LATEX were simply modified to work with the regular *TUGboat* macros. The main impetus for not converting Clark's article was the inclusion of a full-page figure (this figure and others were pasted in from copy prepared on a laser printer; the AMS typesetter was not able to produce them, and besides they were intended as examples of a particular software/hardware combination). This was a simple matter of expedience; the plain-based macros did not have an automatic method for leaving a blank page, while the LATEX macros did. The author helped out by providing the prototype of a style option that would make LATEX's `article` style produce pages that looked like

TUGboat. This style option has since been refined and extended so that it is now nearly impossible to tell which articles have been prepared with LATEX, other than those using the `doc` option.

TUGboat has always been a showcase for new TEX-related software and it's one of the joys of being editor that I often learn about new goodies long before most other TEX users. One of these new goodies was PICTEX, which was introduced in *TUGboat* 9, no. (2). The article appeared one day in my mail pile on an unsolicited diskette. It arrived complete with all the auxiliary software and test files, and even already used the *TUGboat* macros. The only problem was, it wouldn't run within the memory allocation assigned for the local production implementation of TEX, even though that was boosted about to the limit available with a half-word address space. Some severe pruning of the *TUGboat* macros, several hours of phone time talking to the author, and explicit page breaks in a few strategic places permitted the article to run to completion. The report from `\tracingstats` showed that, at the critical page break, only 8 bytes of memory remained untouched. This article has joined the Boston Computer Society benchmark examples in the test suite for the PC benchmark mentioned earlier.

A hint of the future

With Frank Mittelbach's discovery of TEX, or more specifically, of LATEX, the proportion of articles prepared with LATEX, and more particularly with the new LATEX styles—`doc`, `multicol`, the new font scheme, and so forth—has blossomed. In *TUGboat* 1, no. 1(2), the most recent regular issue, more than half the pages, and two thirds of the articles were prepared with LATEX. We expect this growth to continue, and are looking forward to the new generation.

How an issue gets put together

Until 1988, I had attempted to produce as much as possible of the camera copy for an issue in a single TEX run. I felt that this was a demonstration of how TEX was really supposed to be used. It was also a good way of flushing out unintended interactions between one author's macros and the next, and enabled me to post editorial warnings to unwary users. And, most significantly, I didn't yet have a production assistant. As soon as one goes from a solo operation to a cooperative arrangement, procedures have to change.

I should point out that the job of *TUGboat* editor is volunteer. I have a full-time job in the Computer Services Division of the American Mathematical Society, and though that is closely involved with TEX and other composition matters, the production of *TUGboat* is not included in its duties. Some work on *TUGboat* can be done during my regular working hours (and is billed to TUG), but most is done in the evenings and on weekends. In 1988, TUG hired someone knowledgeable in TEX to be the *TUGboat* production assistant. Early in 1989, Ron Whitney, who was formerly the head of the Composition Department at AMS, and with whom I had already worked on several major projects, became the TUG TEXnician.

The division of labor between editor and production assistant isn't always clear. In practice, Ron now prepares the first draft on paper and delivers it to me to read and comment. But if I happen to be checking the mailbox that day, I may get fascinated with an article that has just been submitted and do the preliminaries myself so that I can see it sooner. But both of us read every submission, and pay especial care if it's an article introducing new macros or a new LATEX technique. (After a gap of several years, we've recruited an associate editor for the macro column who should make this part of our job much easier. Victor Eijkhout's byline will appear for the first time in *TUGboat* 1, no. 1(4).) Although the editor has the final say (and should take the blame when something goes wrong), in the present arrangement we both feel responsible for trying to make *TUGboat* a publication we can be proud of.

New macros for TUGboat

During the summer of 1989, Ron and I discussed at length what kind of user interface we really wanted. Needless to say, it would identify logical elements rather than typographic ones. We tried to regularize the syntax, and, for the plain-based macros, adopted some ideas from LATEX, in particular the bracket notation for optional arguments. As LATEX already provides notation for most of the structures that are likely to occur in a *TUGboat* article, all we had to do there was make sure that the formal characteristics of the output conformed to the *TUGboat* style. (It was also necessary to subvert some of the LATEX `article` style conventions to permit multiple articles to be run together, but that affects only the *TUGboat* production staff, not authors.)

Then Ron took apart and reconstructed the plain-based macros to obtain the desired effects, and we wrote an article that not only instructs authors how to use the new macros, but was also the stress test to demonstrate that they worked. Instead of writing a separate set of instructions for the LATEX style option, we included a section in the article that describes the distinctive macros for *TUGboat* top matter, and refers authors to the LATEX manual for everything else.

We now expect authors to use the macros provided. We try to keep an up-to-date version installed at the major archives, and we will send copies on diskette to authors who have no network contact. It doesn't really matter which version an author uses—plain or LATEX; even though we can't mix them in one run, we have devised simple and reliable TEXniques for starting an article anywhere on a page, just by providing the information on

where the previous article ended.

There are still some legitimate reasons for authors to provide camera-ready copy instead of (or in addition to) sending us the files. The most common reasons are

- requirement for special fonts (e.g., Japanese or pointing hands), although we can do and have done extensive font work when METAFONT sources were provided with enough lead time;
- the intention of showing off the quality of the copy produced by a particular output device;
- the inclusion of an illustration that is relevant to the article, but that wasn't prepared with TEX or requires hardware that isn't available to the production staff.

The joints that result from discontinuities between plain and LATEX articles are patched, non-TEX insertions are pasted in, and the final touches (like running heads) on copy provided in camera-ready form are added by hand, with the help of a razor blade and waxing machine. We don't believe that, in most cases, readers can tell the difference between items prepared in one way or another. But we do think it's important to identify the different TEXniques used, so we provide a **Production notes** column in every issue.

Help for authors and readers

As one of our post-production chores, we try to send back to every author the actual file used for the article as published. (We also send a new copy of the style files if they've changed.) When editing the article files, we try to comment out the important bits that we've changed, rather than simply making changes, and to add comments where we think our intentions may not have been clear. Then the author can see what we consider good practice, and might learn from it.

We also think it's very important to point out to both author and reader any special production problems. I've already referred to one such problem, with PICTEX and limited memory.

Another problem that occurs more frequently than we'd like is the redefinition by some authors of either plain control sequences or even TEX primitives. Surprisingly, most primitives aren't protected against redefinition, and even experienced TEXers often don't think about the consequences of what might happen if they choose an "obvious" name for some macro, and it just happens to be the name of a macro that's intimately involved in, say, the output routine. (Try the definition `\def\vbox{\hbox}` and a short paragraph or two.) Most re-uses of macro names aren't quite so obvious, though, and users who pick up some new macros they've just read about without considering the possibility of name conflicts could be in for a big surprise if no warning has been given.

What have we learned?

TUGboat, in order to be a useful publication, must be self-referential. That is, the content of the articles will often describe the techniques used to produce the articles.

We have to keep ahead of the authors

TUGboat authors are highly inventive. For example, TEX wasn't designed to support pictures; that's what the `\special` command is for—it "enables you to make use of special equipment that might be available to you, e.g., for printing books in glorious TEXnicolor." But authors seem to ignore that small restriction, and figure out ways to prepare graphics within TEX, without `\special`s. Even Donald Knuth has ignored his own advice and developed a variety of fonts to be used for halftones (*TUGboat* 8, no. (2)). Nobody has yet submitted an article that requires color printing, but I guess we should expect it. (An article by Ken Yap, *TUGboat* 1, no. 1(2), does tell how to prepare color slides.) And I am quite sure that the day will come when such an article appears in the *TUGboat* mailbox that *doesn't* require `\special` processing but does ask for color.

The moral of this is that the editor had better be prepared for anything. It's probably true that a publication like *TUGboat* can't be edited successfully by someone who is less familiar with the use of TEX than the average author, and the authors are getting a lot more clever.

We have to stay organized

Attention to procedures shouldn't be forgotten. We've learned that communication between members of the production staff is absolutely essential. It's no fun to find that the changes you just made to the grammar in an article have just been obliterated by equally necessary updates to the macros made by the other worker. And it's also a big shock to find that the macros that worked yesterday aren't recognized today, because a new version of the style file has just been installed. Even if it's a one-person operation, it's necessary to keep careful records, so you can identify exactly which of several versions is the one that's to go to press.

Let's not forget our typographic heritage

I'd like to make one final point: TEX users, both authors and readers of *TUGboat*, may not be familiar with the traditions of fine typography. We have a tool that is capable of being used with great precision and craftsmanship, if only we know what models to follow. It is one of my goals to try to educate this audience, myself included, in the best and most appropriate ways to use this tool. The editors of the Seybold Report on Publishing Systems have stated the case well:[2]

[2] Seybold Report on Publishing Systems, Vol. 19, No. 22, August 20, 1990.

Over the years, in spite of the remarkable productivity gains that it has produced, the computer revolution has left its mark in a negative way: the craftsmanship that went into certain parts of composing type has been sacrificed. We're not suggesting that we go back to the old ways, but we are making a plea toward slowly raising the current standard through increasing awareness of the issues and opportunities.

BIJLAGE T

SGML and TEX at Elsevier Science Publishers

Jeroen Soutberg
Elsevier Science Publishers
Amsterdam, The Netherlands

Augustus 31, 1990

Contents

Introduction

SGML and TEX are in use at Elsevier as production tools for an increasing variety of products. In the following we describe the prevailing trend in publication technology, and the roles SGML and TEX play in the developments at Elsevier.

ESP

Elsevier Science Publishers is one of the larger scientific publishers in the world. We publish about 600 journals and annually about 600 books in several fields of science. The publications appear under two imprints: Elsevier and North-Holland.

Elsevier

The Elsevier imprint is used for publications in biology/medicine, chemistry, agricultural and earth sciences. Well-known examples are:

- Biochimica & Biophysica Acta,
- Journal of Chromatography,
- Ecosystems of the World (book series).

North-Holland

The North-Holland imprint is used for publications in physics, materials science, engineering & design and mathematics and linguistics. Examples are:

- Nuclear Physics (A, B),
- Surface Science.

Manuscript routing

The publishing world is in transit from a state of established procedures and practices to a state of constant adjustment to the technical and procedural opportunities arising from technological developments. These changes have a profound effect on the routing of manuscripts in the publication process.

We give a simplified description of this transition by splitting it up into three stages:

- The traditional situation;
- The present – transitional – situation;
- The future: database publishing.

Manuscript routing – traditional

Traditionally there have been two ways of producing a printed publication: (1) Preparing a typeset publication from the manuscript submitted by the author, and (2) printing a publication from camera-ready material prepared by the author.

These scenarios yielded quite different print qualities, but correspondingly different costs were involved. These differences made it relatively easy to choose a scenario for a project, depending on its specific requirements.

The traditional scenarios are shown schematically in figure 1.

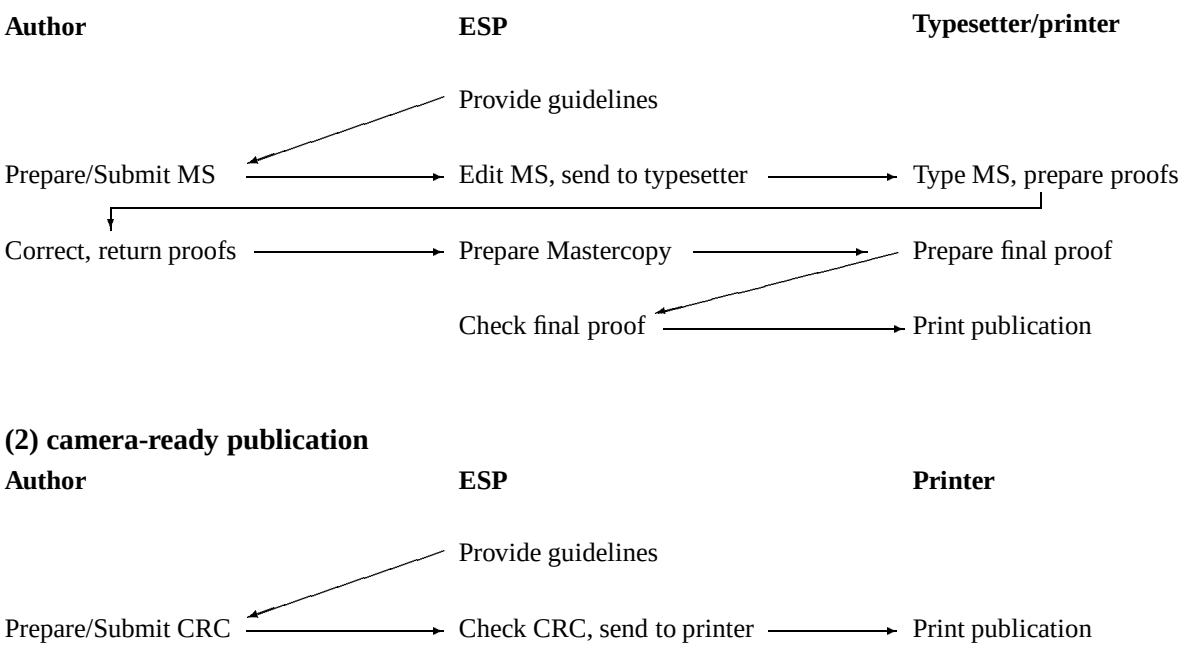


Figure 1: Traditional scenarios for printed publications

Manuscript routing – present

The advent of electronic text processing (in typesetting systems as well as in 'word processors') and the availability of laser printers has upset the clear distinction between the traditional scenarios.

On the one hand, the CRC output an author may produce using a text processing system or a DTP package is much better than in the past, and it may even approach the quality of typeset material. On the other hand both authors of traditional manuscripts and authors of CRC material may want to submit their information in electronic form, in the first case for saving proof-reading time or accelerate publication, in the latter because they do not have a high-quality output device available. In both cases the publisher will have to process an electronic 'manuscript', possibly with media and/or code conversion. Also, the boundary between publisher and typesetter becomes less clear (who does the conversion, who edits the files, ...).

Another trend in publication is the preparation of secondary information from primary information and the re-use of primary information in new products. It is desirable for these purposes to have the primary information available in electronic form.

Figure 2 shows the schematics for typeset and CRC material with the conversion stage and re-use taken into account. In comparison with figure 1 the distinction between typeset and CRC material is much less clear; theoretically it is now even possible that a typeset and a CRC publication end up following the same routing. Also, these are not the only products anymore.

Manuscript (information) routing – future

Extrapolation of the trends sketched above leads to a situation where information reaches the publisher in many different formats, and is processed into many different end products. The simple distinction between typeset and CRC material then loses its relevance.

It is not feasible to construct bridges ('translation programs') between all of these input and output formats. Apart from the number of programs required (number of input formats $\times$ number of output formats) the need to store information for future re-use would imply storage facilities for the same number of formats.

The ultimate solution to these problems is the concept of 'database publishing': the publisher has all incoming information converted into *one* intermediate format; the information is stored in a database in this format, and all end products are manufactured by making a selection from the database and converting this information into the appropriate format. In this way we have one standard storage format, and the number of translation programs needed reduces to the sum of the number of input and output formats. Figure 3 presents a schematic picture of this concept.

The publisher has now become an 'information broker'

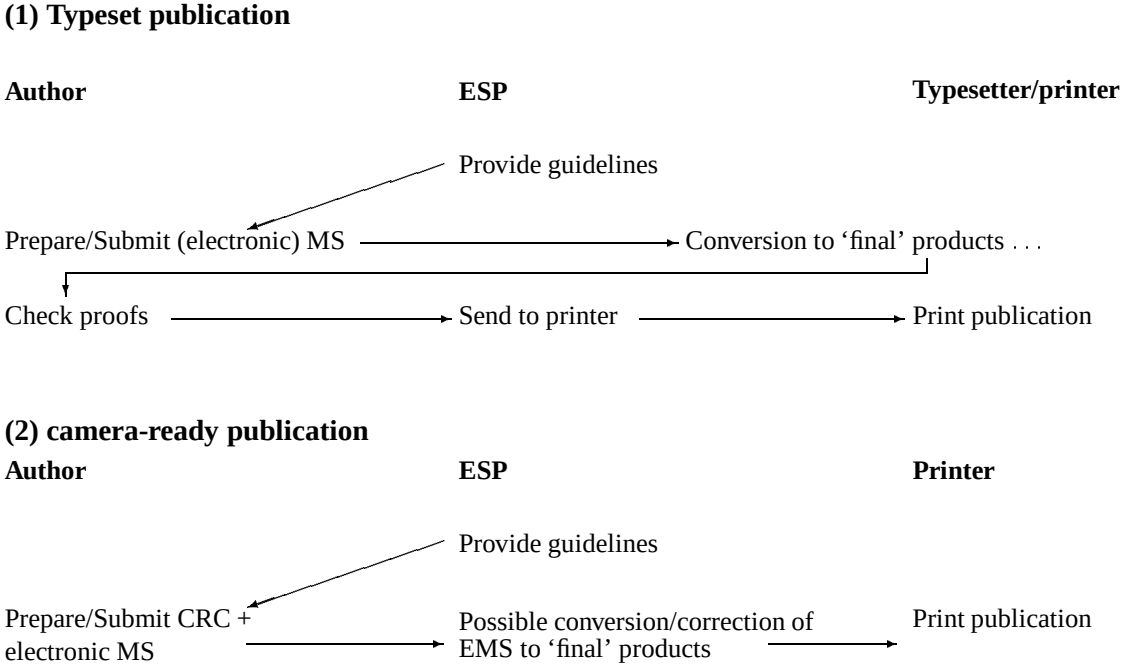


Figure 2: Transitional scenarios for printed publications

with all its valuable material stored in the database. Correspondingly, an author now submits information for the database.

The fact that many different outputs are to be generated from the intermediate format poses strict requirements for this format:

- It should be *presentation-independent* (e.g., the possibilities to identify a heading by its presentation are quite different for printed publications and on-line applications, so the database should contain the heading identified as such; not less, not more).
- It should be *consistent and clearly defined* (e.g., every identification should identify the same, well-defined piece of information).
- It should be *internationally accepted* (to make it possible to exchange information with others).
- It should be *implementable* (e.g., software and hardware available).

The requirements for the intermediate format affect the instructions to authors as well: now the material produced should be in a format which contains sufficient information to generate the intermediate format. This could mean, for instance, that references should be submitted in a special format which can be converted automatically for the database.

Role of SGML – the intermediate format

Over the last few years SGML has emerged as the primary candidate for storing information in the manner described in the previous section. It goes a long way to meeting the requirements for the intermediate format introduced there. In a different order than above the requirements are met as follows:

- SGML is internationally accepted, an ISO standard.
- SGML is implemented in processing software like Softquad, Writer Station, et c.
- To a large extent, SGML is consistent and clearly defined; over the last year we have found some problems which we dealt with by creating preferred versions of ambiguous solutions for use within Elsevier.
- SGML is meant to be presentation independent. In the last few years a discussion has been going on within Elsevier about the question how far one should go in this. For example, the simple equation $s = v_0 t + \frac{1}{2} a t^2$ cannot be presented in presentation-independent form unless the entire formula is spelled out. Even then it will be difficult to make the factor $\frac{1}{2}$ presentation-independent. The same problem arises in all instances where *for reasons of efficiency* a special presentation was introduced for identification purposes. Our present projects adhere to presentation independence down to the level of the *structure* of the information.

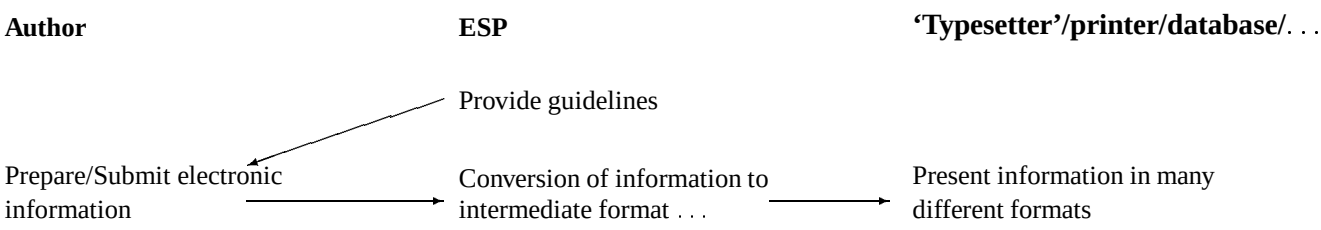


Figure 3: Future scenario for information processing

Role of TEX– Input format, output format

At Elsevier, TEX is used mainly for two rather different classes of applications:

- Where SGML is used as the intermediate format we use TEX as (one of the) output facilities. This permits us to generate, among others, high-quality paper output of proof material formatted in such a way that the information loaded into a database can be validated without looking at SGML codes.
- TEX is also used as the combined input–output format in projects where compuscripts from authors are processed in TEX all the way through. The conversion of TEX to other formats – which needs considerable manual correction – is thus avoided. Also, high-quality output can be generated locally.

Projects

As examples of the different classes of applications mentioned in the previous section we describe two ESP projects representing these classes

CAPCAS

The abbreviation, CAPCAS, stands for Computer-Aided Production of Current-Awareness Services. Although the system can do more or less what its name implies, its status has changed with developments. It is now intended to be a first step towards database publishing.

The scope of CAPCAS is at present limited to the storage of the Heads of journal articles in a database. (By Head we mean the bibliographical information and the abstract at the beginning of an article). Article tails and bodies will be dealt with later. As explained above, the information in the database is stored in SGML format.

Input is provided by our typesetters. This is at present the easiest input route, because a vast majority of articles are still being submitted on paper and the information becomes first available electronically after key-in by the typesetter. The typesetter prepares both the files for the typesetting equipment and the SGML files for CAPCAS from the key-in.

Output is generated for in-house and external applications: in-house for validation of information loaded into the database, contents lists and indexes, externally for secondary services.

- The validation of information loaded into the database concerns mainly the correct identification of fields in the Head, especially those which do not show up in the printed article (e.g., city and country are not typographically distinguished from the organization name in the affiliation). After loading the Head of an article into the database a so-called 'load report' is produced using TEX, whereby all the SGML tags generate unique layout features. As a result, the fields may be identified by simply looking at the layout instead of checking the positions of tags.
- Contents lists and indexes are generated from the database in SGML format. One of the present options is to produce output with the help of TEX, making use of the typical TEX 'programming' possibilities for handling information.
- As yet, no products are being delivered to external customers: we only just started loading the database. Also, before any products may be delivered the exact specification has to be established in consultation with the customer.

Future developments will be the incorporation of the other constituents of articles: tails (i.e. reference/note section), and (ultimately) the bodies (main text, including formulae, tables, figures).

We will start working on the tails fairly soon (probably in the coming year); the body, by sheer size and complexity, will not be tackled for some time to come.

LATEX compuscript processing

Similarly to other publishers, there are several Elsevier journals where most manuscripts received are coded in TEX or LATEX. A project has been set up to create an article routing in which the electronic version of the manuscript is processed, keeping within TEX, and the output is produced with TEX as well; proofs are printed on a laser printer while the final output is produced in high resolution on the in-house typesetting machine. Special software makes it possible to generate output using 'Times' fonts instead of cm fonts.

BIJLAGE U

NTG's second year

C.G. van der Laan

May 1990

Organizational

NTG (Nederlandstalige TEX Gebruikersgroep), the Dutch language-oriented oriented TEX Users' Group has existed inofficially since June 1988, and officially since the fall of 1989. The Group currently has 18 institutional members (representing 54 individuals) and some 50 individual members, making the total roughly one hundred members.
Members of the TEX Users' Group regularly receive minutes of the spring and fall meetings, with articles and status reports collected in the appendices, as well as the most recent updates of the membership database. At the meetings, organizational aspects (such as the budget) are discussed. Presentations are also given. Most recently, these included 'Unusual paragraph shapes' by Victor Eijkhout, 'SGML and TeX at Elsevier' by Nico Poppelier and 'Various network aspects' by Johannes Braams. Roughly 40 members are present at each such meeting.

The listserver `tex-nl@hearn` is heavily used for information exchange and asking questions. The fileserver `tex-nl@hearn` contains codes useful to the Dutch-speaking community. The University of Utrecht (the RUU for short) recently made a second (internet) fileserver available to this community. Those wishing to contact the board of NTG can also do it via email number: `ntg@hearn`.

The NTG's first year was characterized by 'getting started' and the second by 'getting organized'. We now face the difficult task of establishing NTG continuity: that is, to get a renewal process going for the election of (two) board members every year, their job being to preserve and stimulate an active NTG. Of course, the real basis for the NTG's life and existence are the Working Groups and individual research. Thus, these activities should continue to be stimulated. Nevertheless, it is important that a structure be established which will continue to provide for 'teaching the teachers'. The TEX Users' Group hopes that also its contacts with other European and international TEX Users' Groups will endorse and strengthen this continuity.

Cooperation with other (European) TEX Users Groups has resulted in:

1. a representative of other Users Groups being invited to the (open) meeting(s) at no (conference and one course) costs,
2. the president of every national or linguistic TEX Users Group being vice-president of the board of directors of TUG (one federative world TEX Users Group, decentralised nationally and linguistically, is strived for),
3. the secretary of every TEX Users Group being member of the other User Groups by default (information exchange is organized),
4. members of one group being considered members of other groups with respect to admission fees for meetings,
5. sharing of know-how (exchange of teachers, speakers etc.).

With respect to TUG, we welcome the realization of the possibility of joint memberships.

Cooperation with the SGML Holland Users' Group has resulted in the SGML-TEX conference, 31 August 1990, and a shared working group. Members of this working group have worked on SGML and TEX with respect to mathematics (C.G. van der Laan of the University of Groningen and D.C. Coleman of Elsevier Science Publishers; J. Grootenhuis of Circe BV, presented a paper at Markup90 in Charleston) and tabular material.

Working group activities

Although we mainly operate in working groups — which help structure discussion and stimulate cooperation — much work has been done on an individual basis. Examples of the latter are the articles 'Unusual paragraph shapes', and 'Typesetting bridge via (plain) TEX,' which have recently been published in TUGboat.

Education

A survey of (local) courses and courseware is maintained. Tools for preparing and maintaining (course) transparencies are in preparation. A set-up of courses has been made for the 'NTG Days,' keeping the idea of worldwide modules in mind. A contribution to Child's discussion of what every module should contain, has been made. Dutch members also teach at other TUG meetings. (E.g. Kees van der Laan held the SGML class (Stanford89), and Victor Eijkhout will give the TEX advanced course at Cork.)

Guidelines for authors

A report (*Journal Style Guidelines*), RC RUG Report 26, has appeared.

PCs

Some evaluation of public domain (PD) MS-DOS programs have been made. An Atari PD set is available upon request. It looks like that this WG will become more active this year, which is very much appreciated. Cooperation with other user groups is bound to be beneficial.

Fonts

A chess font has been developed by Tutelaers (to be published in TUGboat).

NTG Days

The first 'NTG Days,' June 89, was organized by RUU. Roughly 80 people attended. This year the theme is SGML and TEX. It was organized by NTG and SGML Holland. The number of courses, offered during these NTG days, has been increased from 2 to 8, where 5 courses have actually taken place. We now have multi-day courses, as opposed to the one-day courses last year. This year's meeting will be on the 31^{st} of August, with courses being held in the week before and after.

SGML(, TEX ...)

An introductory paper was prepared and presented at the second SGML Holland seminar (October 89, Amsterdam), and at GUTenberg 90 (May 90, Toulouse). The final version will be presented at Cork90 and the fall NTG meeting (see appendix —SGML (, TEX and ...)···)— for the printed version.)

Dutch aspects

Various sty.files have emerged for `report`, `article` and `letter`. In addition, articles have been prepared on A4, international LATEX, and Babel. These articles have been submitted to TUGboat.

Communication

Fileservers — `tex-nl@hearn` and RUU's internet server — have been maintained. Problems of what and how (coded?) to store, and how to retrieve, are under study. A floppy distribution service has been started: floppies with information (minutes with appendices) as well as programs (those available on the fileservers) are available, especially to those members who don't have access to electronic mail.

Some opinions

In the last meeting the NTG was urged to pay more attention to Public Relation activities, e.g.: 'helicopter presentations' such as 'What is TEX all about (off the shelf),' and a real survey course on: TEX, LATEX, Metafont, Postscript, SGML and how these are related.

The idea of an NTG-report series has been opted.

Furthermore, attention will be paid to Public Domain PC versions, not so much to develop it ourselves but to get the material, exercise it, and organize dissemination.

BIJLAGE V

Development of DANTE e.V.

Joachim Lammarsch
Computer Center, University of Heidelberg, Germany
(X92@DDURZ1)

June 1990

Introduction

Reading the material I got via distribution mail from TUGboard and the minutes from finance committee I stated some correspondence between TUG with about 4000 members and DANTE e.V. with about 500 members (after 1 year of its existence). During the last several months there has been the question for DANTE e.V., as there has been for TUG, as to which direction the development of the group should go.

Spectrum of users

Who will and shall use TEX/LATEX? If you limit the use to universities, the group of users is small and will decrease more and more, because more and more commercial software with a lot of advertising will set the direction of application for users. That means that you may not wait till the users will come to the university because in most cases then it's to late. A logical consequence is that you have to spread out TEX before students come to university, thus in high schools. Otherwise there is the risk that the current users will die out little by little and no new users will come up. But if a lot of students know TEX/LaTeX from the high schools and universities they will take it with them if they are going to the commercial world.

TEX in high school

There are two reasons why TEX is interesting for schools in Germany. First of all money is the most important factor. German schools have only a small budget for soft- and hardware. So if they can save money by not buying software, they have more money for hardware.

Teachers are interested in TEX for typesetting their papers for education in a good quality (as you may see by the teachers who are members from DANTE e.V.). The problem teaching teachers has been solved by DANTE e.V. in that way, that experienced members agree to give a free introduction to TEX/LATEX in high schools. Thus trained teachers may distribute TEX/LATEX just by using it.

There are as yet no plans to teach TEX/LATEX to students. But in some high schools workshops to students. But in some highschools workshops exist, which use typesetting systems like Word Perfect — why not LATEX?

TEX for the youth

Since TEX is public domain for PCs a lot of persons are interested in it. But for these persons the problem is to get education. That problem may be solved by giving courses for beginners as DANTE e.V. is doing twice a year without fee. Another solution could be to offer courses at night-schools. That is in Germany a payable alternative to high school and university.

TEX in journals

In Germany there are a lot of computing journals which partly have a very high circulation. We should try to get regular columns there to present and discuss TEX/LATEX and related themes to a big audience. Also there is in Germany is one journal where such a column already exists. With another one, DANTE e.V. has contacts and plans to establish such a column.

TEX in TV

By chance some days ago I met the person who is responsible for a series on TV about computer science. This is another area where DANTE e.V. will try to gain a foothold.

Result

It's not necessary to make advertising for TEX and related software like for a soap or a car or TV. But you have to look for the needs of users and fulfil these needs. You have to decide what you want to be: one of a small group of persons who know, understand and use TEX as a wizard (and partly earn a lot of money). Or to be one of a crowd who use and spread TEX. What is better for TEX, its development or the groups which support TEX?

What should be done?

Everyone wants to have the ability to get a version of TEX quick and easily. DANTE e.V. has a listserv on Bitnet, an FTP server on Internet and a mailbox. Besides DANTE e.V. is distributing software via diskettes for free to its members.

Information about what's going on in the TEX world and information for beginners must be available for everyone. DANTE e.V. is publishing quarterly a journal. For members with email access three mailing lists exist: `TEX-D-L` in general for TEX, `TEX-D-PC` for questions concerning TEX on PC, and `TEX-EURO` for support of users in Europe. All the information on these lists is available via diskettes for members who have no email access.

Everyone should have the ability to get immediate help for a problem. Members can call DANTE e.V. for that, and if it's possible the problem will be solved at once. Otherwise the question will be refered to a group of specialists. The second way is for members to ask directly one of the side-coordinators from DANTE e.V. There are for the most important systems and areas side-coordinators in DANTE e.V.

What's about the money?

Organization costs (normally) money! But it's not entirely true. Until now DANTE e.V. was able to manage all its business without own staff. It was possible to find volunteers who are doing all the work.

DANTE e.V. has the support from computing centers (University of Heidelberg, MPI Stuttgart, University of Düsseldorf, University of Stuttgart) and does not have to pay for the use of the mainframe. The mailbox has been made available by the company Breakpoint GmbH (Holzkirchen/Germany) for free.

So it has been turned out that you have either to pay for things or you have to treat to get things for free. DANTE e.V. has got a lot of things for free by negotiation!

BIJLAGE W

GUTenberg '90
Thema: TEX en de Grafische Kunst

C.G. van der Laan

15–17 mei 1990

TEX3.x is essentially about fonts

Samenvatting
- Zusterclubs zijn automatisch lid van elkaar.
- DANTE en GUTenberg 100% gegroeid.
- TUGlib wordt uitgebreid met TEX bibliografie.
Producten/macros
- Zetten commutator diagrammen via LATEX.
- Ook muziek zetten nu via TEX.
- Zetten van (ADA, . . .) programmas.
- Zetten van bridge literatuur.
- Voor multilingual TEX3.0 is MLTEX nog nodig.
- TEX in kleur?

1 Algemeen

Als NTG vertegenwoordiger hoefde ik geen deelnamekosten te betalen. De proceedings —GUTenberg cahiers 5— en locale informatie werden aan het begin van de conferentie aan de deelnemers uitgereikt.

Het programma bestond uit:
- ma 14 mei: TEX-EOB vergadering,
- di 15 mei: Workshops en LATEX introductie cursus (parallel),
- wo/do 16-17 mei: Ledenvergadering en de eigenlijke conferentie,
- vr 18 mei: Cursus plain TEX macros (Seroul).

Ik had deelgenomen aan de TEX-EOB vergadering, een workshop en uiteraard de conferentie. Ik heb mijn zegje gedaan in de EOB-vergadering en twee verhalen gehouden: TEX and SGML, en Typesetting bridge via plain TEX.[1]

2 Reizen etc.

's Morgens 13 mei vertrokken naar Parijs. Vandaar aan het eind van de dag doorgereisd naar Toulouse. Donderdagavond 17 mei de nachttrein naar Parijs en de volgende dag doorgereisd naar Garnwerd. Het traject Parijs ⇔ Toulouse heb ik per couchette afgelegd. In Toulouse heb ik, om de kosten te drukken, een goedkoop hotelletje opgezocht. (Hierdoor vermijd je de registratie overhead, en bovendien kun je goedkope hotelletjes niet boeken via de congresorganisatie of 'VVV').

3 TEX-EOB

Deelnemers aan de vergadering: Bernard Gaulle (gastheer en voorzitter), Nelson Beebe (TUG president), Malcolm Clarke (Europese coordinator), Joachim Lammarsch en Luzia Dietsche (DANTE), Philippe Louarn (Editor Cahiers), Olivier Nicole (LATEX docent), nog enige GUTenberg bestuursleden en later Jacques André. Er zijn geen notulen gemaakt.

Wat is er allemaal gebeurd? Vorig jaar in Parijs was de eerste bijeenkomst, vervolgens de happening in Stanford, en, informeler, het weerzien in Karlsruhe, waar o.a. Cork is voorbereid.
Gespreksonderwerpen:
- **Vice-presidents TUG useful?**
  Voorzitters nationale/taal-groeperingen zijn vicepresident van TUG (Het nut hiervan wordt m.n. door Bernard erg in twijfel getrokken; ik vind het wel nuttig o.a. doordat de informatiestromen nu structureel geregeld zijn van TUG naar ons. Omgekeerd is het de bedoeling dat wij als TUGboard leden meedenken

[1] Werk in 'progress.' Het bridgeverhaal, in definitieve vorm, verschijnt in TUGboat 11#2. TEX and SGML zal verder uitgewerkt worden – met tabellen en linking — voor de Europese TUG-bijeenkomst in Cork en de NTG najaarsbijeenkomst.

over de TEX Users Groups evolutie. De meningen lagen verdeeld: DANTE en GUTenberg zijn het met elkaar eens, terwijl Malcolm en ik meer op een lijn zitten, een meer cooperatieve attitude hebben. Bernard en Joachim willen meer naar zich toetrekken.)

- **Voordeel leden t.a.v. zustervereningingen.**
  Leden van een zustervereniging kunnen deelnemen aan de andere bijeenkomsten tegen ledenprijs. (Dit punt was vorig jaar aangestipt en inmiddels de facto door de Europese TEX Gebruikersgroepen geadopteerd. De discussie draait om de houding van TUG jegens de overige groepen.[2] Hierdoor heen loopt het zwevende wederzijdse lidmaatschap. Nelson zou deze beide punten meenemen. De relevantie naar de andere Europese clubs is niet zo groot: in Karlsruhe waren 4 Nederlanders, nauwelijks Fransen (toegegeven het was niet door DANTE georganiseerd); in Parijs was ik de enige Nederlander; idem in Stanford; nu was er nog een Nederlander, n.b. (nog) geen NTG lid; en in Engeland, dit voorjaar, nam Joop van Gent deel, toen wel lid van ukTEXug en (nog) geen lid van NTG. Het effect van deze maatregel is vooral psychologisch. T.a.v. de Europese TUG bijeenkomsten acht ik het meer relevant.
- **Informatie uitwisseling.**
  De zustervereningingen zijn automatisch lid van elkaar; de Europese coordinator is 'lid' van elke TUG. Dit punt is hier besloten en garandeert o.a. de uitwisseling van informatie.
- **TEX-EOB bijeenkomsten.**
  Ook vonden wij elkaar dit jaar eindelijk in de vergaderfrequentie en -plaats: eenmaal per jaar, en wel op de Europese TUG bijeenkomst, dus in Cork dit jaar. Daarnaast is het natuurlijk helemaal niet verboden om met elkaar contact te hebben en eventueel zaken voor te bereiden.
- **Stand van zaken.**
  Wij begonnen trouwens met het schetsen van elkaars wel en wee sinds vorig jaar.
  Malcolm: ukTEXug heeft nog geen officiele naam. Zijn bestuur ontbeert daadkracht vanwege het staken der stemmen. Hun bijeenkomsten krijgen steeds meer het karakter van een workshop, de laatste was over bibliografieën en indices. De Exeter proceedings zijn uit.
  Kees: zie mijn 'jaarverslag',[3]
  Joachim: DANTE's ledental is verdubbeld.
  Joachim deelde 'Die TEXnische Komödie 1,' uit.[4]
  Cursussen tijdens de DANTE bijeenkomsten zijn zo goed als gratis! (Malcolm en ik vonden dat dat structureel niet vol te houden is en dat het zondigt tegen het principe 'de gebruiker betaalt.' Nu zijn er oneigenlijke geldstromen of vreselijk enthousiaste leden die gratuite willen werken.)
  Gewezen werd op Breitenholers benadering van het meertaligheidsprobleem ('chapter'-probleem). Zijn artikel is of zal verschijnen in TUGboat.
  Overigens, zoals bekend, verzorgt Luzia de Duitse samenvattingen van TUGboat.
  DANTE krijgt er ook een internet server in Stuttgart bij (150 à 200 MByte).
  PR activiteiten via o.a. artikelen in tijdschriften.
  (Een subtiel onderscheid werd gemaakt tussen DANTE leden en de Duitse TEX geïnteresseerden. Tot de laatste categorie behoren o.a. Frank Mittelbach, Rainer Schöpf en Anne Brüggeman-Klein. Stefan von Bechtolsheim?).
  T.a.v. TEX op middelbare scholen was geen nieuws te melden.
  Bernard: GUTenberg is ook drastisch gegroeid, 350 leden nu.
  Zij werken samen met DANTE m.b.t. de fileserver.
  Telkens terugkerende wekzaamheden, nog steeds gedaan door vrijwilligers, zijn: secretariaat GUTenberg, en listserver onderhoud/reviewing, productie cahiers.
  GUTenberg cahiers: 4-maal per jaar.
  Ook zij distribueren PD PC versies. Daarnaast verspreiden zij Ferguson's MLTEX. Een floppy service dus.
  Zij zoeken vrijwilligers voor de redactie van de cahiers en vrijwilligers die als Goeroe, lees vraagbaak, willen optreden.
- **Organisatie.**
  Herstructurering TUG-commissies. Als voorbeeld werd aangehaald: Michael Ferguson's 'Foreign language' volunteering. Er zijn kennelijk twee stromingen. De eerste die zegt (waartoe Nelson, Malcolm en ik behoren) ha prima iemand wil zich ergens voor inzetten, hear hear, en laten wij deze energie zo goed mogelijk benutten, en de tweede die zegt dat er eerst gesproken moet worden over wat gewenst wordt, dat dat vervolgens formeel besloten moet worden (met taakstelling en zo) en dat er dan een commissie verkozen en benoemd moet worden. De angel in dit soort discussies is niet zozeer wie er gelijk heeft — beide werkwijzen hebben hun voor en tegen— maar dat het zo polariserend werkt. Het is veeleer de kunst het enthousiasme levend te houden, verstandig met de energie om te gaan, en die formele structuur te gebruiken die gegeven de omstandigheden het beste is. (Jawel hoor een mondvol, maar hoe doe je dat, hoor ik diverse mensen al zeggen. Welnu, het is dé opgave

[2] Inmiddels is gebleken dat voor de Cork bijeenkomst Europese gebruikersgroepsleden dezelfde korting genieten als TUG leden.

[3] NTG's second year.

[4] Het begint al op een tijdschrift te lijken: Knuth's 3.0 verhaal is er in opgenomen; BibTEX en makeindex bespreking; regel.sty (o.a. voor programma teksten); struct.sty (structuurdiagrammen voor programmeren); TEXPART.BAT (meertalige LATEX documenten); EXPDLIST (een uitbreiding van descriptor omgeving); vergelijking van diverse TEX programmatuur, o.a. de PD versies; MID's The Publisher; diverse aankondigingen, o.a. onze aug conferentie; samenvattingen TUGboat artikelen; en enkele speelse aardigheidjes, zoals TEX-poezie.

van de diverse boards daar goed mee om te gaan.)
Ook werd het verkooppunt van TUG gadgets in Europa bekritiseerd (m.n. dat TUG zonder overleg met betrokkenen, i.e. Europeanen, Peter Abbott daarvoor heeft aangezocht.) Ook hier zit weer wat in, maar ik ontkom niet aan de indruk dat de klagers niet naar het resultaat kijken maar eerder dat zij op zijn minst een vinger in de pap willen hebben. Ik sta achter ReG's benadering om vanaf een of ander betrouwbaar punt in Europa, een set verkoopartikelen langs de diverse bijeenkomsten te laten 'gaan.' Het probleem is echter vrijwilligers te vinden die dit gedegen willen behartigen. (N.b. TUG gebruikt zijn 'personnel' —de betaalde krachten en hun aanhang— hiervoor.)
ReG's 'summit' agenda (Met onderwerpen: Reciprocal membership with TUG, distribution points in Europe for publications and TUG products, helping TEXies in eastern Europe, . . .) werd eveneens bekritiseerd, alhoewel duidelijk was aangegeven dat men agendapunten kan indienen.

- **Wandelgangen praat.**
  Malcolm en ik bespraken tijdens een lunch dat het helemaal zo gek niet is te denken aan continentale TUGboats. Waarom zou GUTenberg niet eens een TUGboat kunnen 'maken.' Tenslotte hebben zij een productieproces tot hun beschikking. Bijvoorbeeld een special issue over meertaligheidsaspecten? Nelson zou ook dit meenemen.
  In discussie met Nelson over de relatie TEX-diroff vertelde hij mij dat het grote probleem met de . . . roff programmatuur is dat er noch kennis noch documentatie over de kern is —de auteur ervan is om het leven gekomen bij een verkeersongeluk.

Nelson had enkele velletjes met notities gemaakt, waarvan hij wel degelijk nota zou nemen, en proberen het beste ervan te maken! Ik heb daar wel vertrouwen in, enne . . . natuurlijk zijn Keulen en Aken niet op een dag gebouwd. Overigens TUG heeft 3500 leden en de diverse Europese groeperingen hebben er zo'n 1000.

4 Dinsdag 15 mei

Tutorials

De overzichtsverhalen die ik volgde waren erg verbaal, en weinig ondersteund met dias of transparanten. Ondanks de simultane vertalingen —erg duur overigens— ben ik niet gewend de essenties van een verhaal uit zo'n stortvloed van woorden op te pikken. Overigens kwam men niet veel verder dan wat voorbeelden te laten zien en dat het toch eigenlijk allemaal erg complex en een kwestie van smaak was.

Workshop: Profesionele kwaliteit bij het publiceren met PC's

Om de sfeer van de workshop enigszins aan te geven de volgende anecdote. Bij meerkolomsuitvoer viel mij op dat de docenten geen uitlijning hanteerden. Ik vroeg of zij dit als beter kwalificeerden. Zij lieten mij vol trots zien dat als je uitlijnt je dan lelijke en onregelmatige woordafstanden kreeg (met hun XPress tool). Ik was sprakeloos vanwege de verwarring van de gewenste kwaliteit met het haalbare resultaat van het gehanteerde product. Ook inspectie van de proceedings leverde weinig harde criteria op: ja, daar was wel wat teveel wit en de lettertypes van de diverse koppen en subkoppen waren niet harmonieus e.d.

Paneldiscussie: Laugier, le Tallec, Marshall

De paneldiscussie kwam tot geen resultaten noch was de discussie geanimeerd. François Chahuneau, die ik ook al bij Kluwer op de SGML-database bijeenkomst ontmoet had, schetste wel de houding van TEXies die iets moois gerealiseerd hadden: . . . terwijl iedereen zich vergaapt aan het resultaat, verdoezelt de auteur hoeveel 'vallen en opstaan' eraan vooraf gegaan is.

5 De conferentie

Na de algemene ledenvergadering, die ik wederom uit beleefdheid niet heb bijgewoond, opende Bernard de conferentie. Hij resumeerde de activiteiten van het afgelopen jaar, zie zijn relaas tijdens de TEX-EOB discussie op de vorige bladzijde.

Nelson Beebe: TUG forward into the '90s

Een overzichtsverhaal (Niet opgenomen in de proceedings, wel heb ik een kopie van zijn transparanten meegekregen).

Bijeenkomsten:
TUG, TEXas, 18?–20 june (met cursussen er omtoe),
JTUG en NTG/SGNL-Holland, 31 aug (met cursussen er omtoe),
TUG in Europe, Cork, 10–13? sept (met cursussen er omtoe),
DANTE, en 9^e bijeenkomst Duitstalige TEX geïnteresseerden, Göttingen, 10–12 oct,
NTG, DEC Utrecht, 20 nov,
DANTE, en 10^e bijeenkomst Duitstalige TEX geïnteresseerden, Wenen, 20–22 febr.

TEX3.0 en Metafont2.0

De aanleiding tot verandering was het zetten van meertalige documenten via TEX beter mogelijk te maken. (Uni-lingual hyphenation needed to be fixed, with en passant improved hyphenation of ligatures etc.). Bij gebrek, op korte termijn, van 256 character fonts (en zelfs wat waar precies moet komen in deze tabellen, helaas geen standaard, en dus diversiteit c.q. verwarring) hebben wij nog te maken met problemen met woordafbreking voor geaccentueerde woorden, en i.h.a. met

woorden die een commando bevatten. (Zie Ferguson's verhaal.) De goede edities van de boeken zijn: Vol A Computers and Typesetting *ninth printing*, zachte kaft versie TEXbook *Seventeenth printing*.

Samengestelde of virtuele fonts

. . . each character is described by a program in DVI language, allowing positioning, rules and even `\specials`.
Dit maakt het mogelijk om letters met accenten goed te doen, en het lost het probleem op hoe TEX fonts af te beelden op randapparaat specifieke fonts. N.b. Volgens Nelson had implementatie op TEX nivo gemoeten: nu moeten de drivers aangepast worden of er moet een virdvi → dvi geschreven worden.

Metafont naar Postscript

Hij gaf referenties naar werk op dit gebied:
1987 Leslie Carr and Sebastian Rahtz;
1989 Victor Ostromoukhov;
1989 Graham Toal & Graham Asher;
1990 Daniel Berry & Shimon Yanai.

Diverse nieuwtjes

Adobe geeft Type 1 font formaat vrij.
Er is een TUGlib@science.utah.edu server, met faciliteiten analoog aan NETlib, zoals 'who is'. (Andere suggesties: Wat is in welke collectie; Actueel literatuur overzicht (Deze suggesties heb ik met hem besproken, c.q. meegegeven.))
Aanbevolen literatuur: David Buerger(1990?): LATEX for Engineers and scientists; en natuurlijk de boeken van von Bechtolsheim, die nog moeten verschijnen.
BIBTEX1.0 zal hopelijk van de zomer verschijnen.

Ferguson: Het nieuwe tijdperk TEX3.0

Hij gaf aan dat de 90er jaren beheerst zullen worden door:

- meertalige documenten,
- exotische standaard fonts,
- gemakkelijker documentuitwisseling,
- ondersteuning door print industrie.

Nodig zijn: Nieuwe 8-bits fonts (hoelang zullen deze op zich laten wachten?), nieuwe drivers, algemene de facto acceptatie door wijdverbreid gebruik, universele LATEX stylen.
Tekortkomingen TEX3.0:
geen automatische (complexe) `\discretionary`,
geen woordafbreking bij samengestelde symbolen, of algemener woorden met een opdracht erin.
Zijn toevoeging —`\charsublist` in MLTEX, die TEX *multi-lingual* maakt— breekt ook woorden met expliciet geaccentueerde tekens af. MLTEX wordt met PCTEX geleverd; GUTenberg distribueert een apparatuur onafhankelijke versie.

Grimault e.a: Beginnerservaringen met LATEX

Uitermate charmant was de bijdrage van deze dames die geen doekje wonden om enige LATEX ongemakken.
Geconstateerde negatieve ervaringen waren:

- Onbekendheid met wat waar te verkrijgen is.
- Het ontbreken van een LATEX introductie in het Frans.
- PCTEX gaf problemen bij het gebruik van andere fonts en daarom is TEXtures gebruikt.
- Eisen opdrachtgevers kwamen niet overeen met LATEX stylen.

Wensen:

- legenda bij tabellen centreren op tabelbreedte,
- automatische breedtebepaling van tabellen kleiner dan `\textwidth`,
- hbt opties zijn *on*gehoorzaam,
- geen handmatige correctie van weduwen en wezen.

Zij bedienen zich van: voorgedefiniëerde toetsen in de editor, preprocessor voor accenten en speciale tekens.

Laugier: Waarom TEX? Waarom niet eigenlijk?

Ook hier blijkt dat motiverende, c.q. wervende, verhaaltjes als nuttig worden ervaren. Hij benadrukte de vele gebruiksmogelijkheden van TEX: wiskunde, chemie etc., met daarnaast Grieks, Russisch, etc., tabellen, bibliografieën e.d. In hun drukkerij hebben zij TEX-Metafont-Postscript geïntegreerd (Ostromoukhov) en gebruiken zij in een net MacintoshII en IBM PC gekoppeld aan een laserprinter (300dots/inch) en fotozetter (1800dots/inch).

Borceux: Het maken van diagrammen (via LATEX)

Een interessant verhaal over 'diagram' macros voor het maken van commutator, homologie, categorie, algebraische topologie etc. diagrammen. (Een aankondiging van een gebruiker over het bestaan van deze macros is gedaan in TEXHaX van jan 90.) Andere diagram programmatuur is: `catmac` van Barr TEXHaX89, en de macros van Taylor (Imperial College). LamsTEX werd als zodanig niet genoemd. Vanwege schalingsinvariantie is gekozen voor de picture omgeving van LATEX. Bij diagrammen zijn de moeilijkheden: schuine lijnen en de lengte van een pijl bij knopen van verschillende grootte. In deze programmtuur zijn zowel de pijlen als de knopen elementen van een conceptueel array. Er is een Engelstalige gebruikersgids. De macros schijnen al op onze fileserver te staan.

Siebenmann: LAMSTEX

Een vrij uitvoerige bespreking van dit pakket dat de functionaliteit van zowel LATEX als AMSTEX biedt: algemeen formaat, commutator diagrammen, en tabellen.

Alhoewel het een uitgebreid verhaal is, bevat het zo goed als geen concrete voorbeelden. (Terloops werd opgemerkt dat Desarmién en Ostromoukhov de picture omgeving in plain TEX hebben geïmplementeerd.) Voor de tabellen wordt geadviseerd deze apart te maken en deze electronisch in te plakken via dvipaste[5], vanwege lage verwerkingssnelheid en groot geheugenbeslag! Bij commutator diagrammen moet men de knopen annex de bijbehorende pijlen herdefinieren. De knopen zijn elementen van een matrix. De PC versie kost ≈$100,-.[6]

Naudin e.a.: Zetten van ADA programmas

Een goede samenvatting is gegeven in de abstract

> 'In this article, we introduce a tool for typesetting programs written in a programming language with keywords. We propose a style derived from the one used by the WEB system to typeset PASCAL programs. The strength of the set of macros that really do the job lies in the use of the internal mechanisms of TEX in order to recognize the keywords, and thus no complex automation need to be considered. This way, the process is of reasonable efficiency, in contrast with the horrible slowness of a classical method written with TEX.'

Voor de lay-out van een programma is een preprocessor nodig. Gegeven een mooie 'presentatie' lay-out, met gereserveerde woorden in hoofdletters, dan kan m.b.v. de macros het fraai gezet worden: vetgedrukte gereserveerde woorden en de rest via italics. Alhoewel geïllustreerd aan de hand van ADA, is het niet beperkt tot ADA. Let wel de kopij betreft documenten met her en der programmas ter illustratie. Dit is dus wat anders, eenvoudiger doelstelling, dan zetten van een groot programma doorspekt met commentaar als kopij; hiervoor is WEB.

Het verhaal behandelt het algemene aspect van: woord voor woord inlezen van tekst, en het herkennen van een ingelezen woord als een woord uit een verzameling, in dit geval van gereserveerde woorden. De macros beslaan 40 regels met 63 definities van de ADA gereserveerde woorden.
En passant wordt een andere toepassing gegeven n.l. het zetten van tekstpuzzles i.e. in een tekst worden woorden vervangen door spaties en dan moet de gemodificeerde tekst, zowel als het negatief, gezet worden. (N.b. Dit kan nuttig zijn bij het vervaardigen van transparanten waarbij men meerdere lagen over elkaar heen projecteert, b.v. bij gebruik van kleuren.)

De auteurs geven zelf aan waar de macros nog geperfectioneerd kunnen woorden: de afdruk voorzien van regelnummers, en het automatiseren van het bepalen van een goede paginagrens.

Een nuttig, mooi opgebouwd en leerzaam verhaal. Als open punt zou ik willen noemen dat er geen onderscheid wordt gemaakt tussen commentaar (ook via italics) en de echte programmatekst. Verder is niet duidelijk gemaakt wat algemeen als mooie typografie van programmas wordt gehanteerd, anders dan wat Knuth via WEB doet.

Guillopé: Streepjescode

Op zich een interessant probleem: het zetten van streepjescode via TEX. De motivatie was automatisering van de registratie van bibliotheekboeken. Elk boek (de collectie bevat ≈20.000 boeken) moest voorzien worden van stickers met gedrukte streepjescode. Het verhaal is onduidelijk en oppervlakkig. De representatie van de invoer lijkt mij niet gelukkig.

Cérin e.a.: Kleurendruk via 'LaTeX'

Na de constatering dat kleur een parameter is naast contrast en helderheid, die de overdracht van informatie kan ondersteunen, rapporteren de auteurs over hun experimenten met LaTeX, Postscript en het gebruik van kleuren laserprinters. Het verhaal draait om: LaTeX aanpassing c.q. uitbreiding met nieuwe macros, voor gekleurde omlijning (tekst, tabel), tekens in kleur, en kleurverschuiving; de koppeling van Postscript (DVIPS, Rokicki); afdruk via QMS Colorscript Model 10. Het is nog niet erg duidelijk allemaal.

Taupin: MusicTeX

Het verhaal werd indrukwekkend gebracht. De problemen bij het zetten van muziek zijn: speciale symbolen (sleutels, noten, stokken, ...), meerdere partijen (het is dus 2-dimensionaal, en in 'real-time'), toegevoegde notaties (bogen, aanzwellen, afnemen, ...), en natuurlijk ook hier weer de afbrekingen (regel en pagina). MuTeX (Steinbach & Schofer; Jalbert) heeft teveel beperkingen, o.a. één partij en de lineaire benadering. Beperkingen zijn nog: handmatige invoering regel- en paginaovergangen, geen aanzwelling/afname. Alhoewel intrigerend is het mij niet duidelijk hoe ik een envoudige fluitpartij hiermee kan zetten.

Mijn verhalen: 'Bridge' en 'TeX en SGML'

Het TEX-SGML-verhaal is als bijlage bij een NTG-verslag verschenen (zie bewerkte en voorlopge laatste versie als bijlage AA bij dit verslag).
Het bridgeverhaal bevat ook algemene aspecten, m.n. het aan de kaak stellen van de verkeerde attitude om

[5] Een TEX postprocessor, zie HITEX, TUGboat, juli 89.
[6] Niet te verwarren met: AMS–LaTeX van AMS, waarover later meer.

programmeernotaties uit karweibesturingstalen, PASCAL, ADA etc., als zodanig in (La)TEX te implementeren. Functioneel zijn bijv. keyword en optionele parameters reeds aanwezig! Daarnaast is de aanpak relevant: kijk eerst wat gebruikelijk is bij de onderhavige (bridge)typografie en probeer vervolgens TEX daarvoor uit te breiden met macros, zo eenvoudig, robuust en gebruikersvriendelijk mogelijk. En —last but not least— weet van ophouden, geef aan wat bewust achterwege is gelaten.

Conclusie

Al met al een gezellige, leerzame en nuttige bijeenkomst, waarbij ik mijn Frans weer eens wat heb kunnen ophalen.
Enige Franse woorden:

ecran	scherm
commodité d'emploi	gebruikersgemak
fichier	file
guillemets: << en >>	aanhalingstekens
logiciel	programmatuur
lexème	token
messagerie	list server
maison d'édition	uitgever
ordinateur	computer
Publication Assisté par Ordinateur	DTP
pilote	driver
police	font
réseau	netwerk

De conferentie op zich was prima verzorgd, evenals 'het uitstapje' naar Carcassonne. De documentie er omtoe echter bevatte diverse onvolkomenheden: verkeerde spelling van namen, onjuiste dan wel onvolledige adresgegevens, cahiers met typografische 'blooper's: her en der teveel witruimte, legenda labelwoorden in het Engels terwijl artikel in het Frans zijn (het 'chapter'-probleem).

BIJLAGE X

Board-of-Directors and Euro-Summit at Cork90

C.G. van der Laan

september 1990

Cork90: the end of a decennium, the start of a new era

Summary

TUG organizational:

- TUG is really open now.
- Working Groups will be sent off with better task descriptions and reporting time schedules.
- How to fulfil better membership needs is proposed.
- A scholarly TUGboat and a separate newsletter is proposed. Topical TUGboat issues are under consideration.
- No increased membership/postage fees for those outside USA.
- The dangling reciprocal membership question is near to a pilot study.
- BoD can operate faster because of adoption of voting by e-mail.
- An extended membership directory is asked for.
- A resource directory is asked for.
- Maintenance/evolution of TEX etc. software must be handled by TUG and LUG s; DEK and LL are out of it!
- More attention will be paid to PR activities: Welcome c.q. Information/Demo packets will be prepared.

Euro-Summit:

- Eastern and Western European countries will exchange information.
- A status report, Euro-Summit Cork90(?), will appear with contact addresses and status reports.

Cork90

Cork is a friendly city by the sea with many bridges —not as famous as the Koningsberger bridges though— with in the past George Boole laying down computer science basics at this very University 1849–64: Boolean algebra. The conference was well organized, with the novelty of e-mail for every participant (Well-done Peter!)

This first TUG conference in Europe must not be confused with the previous European meetings sponsored by TUG, like e.g. the last Karlsruhe meeting. The organization had the usual set-up: conference over several days with one stream of presentations, the vendor booths, board meetings at 'morning, noon and evening,' with courses at the days before and after. The Europeans had their marathon too: The Euro-Summit!

Another novelty was the availability of the proceedings of the TEXas AM meeting. (Congratulations Lincoln!)

1 Board of Directors meetings

E-mail clashes and heavy phone-calls preluded what might be called a phoenix meeting. The key-issue was: How should TUG fulfil its international role, despite its USA roots and lacking time to handle things appropriately. Waves coming ashore from Western Europe, immediately followed by the even higher Eastern European seas, not to mention the waves from the Pacific, especially Japan! Some numbers: TUG ≈3,500 members, Western Europe ≈1,500 members, Eastern Europe 500–1000, with an enormous potential, Japan ≈500 members, Australia ?, China ?, ... One thing for sure: the sea will eventually calm down, let us make a guess at 10,000–15,000 members, within 5 years (another guess), with a tenfold or so out there, just silently running their TEX engine, without much ado. So, how is TUG going to address the international challenge?

1.1 Openness

First, the basics, what about openness? At this meeting the board was keen on *acting as a board* and therefore motioned to have access to all information, especially that privileged to the executive committee. It was also agreed to have the minutes available within a month after the meeting and have these accepted and available for free within another month, with the TUG office nagging. In TUGboat, or the newsletter, the president will summarize as soon as possible the main issues treated, in easy going prose.

1.2 Working groups

Second, the working groups (WG s) need a proper description of their task and reporting time schedules. More

or less so motioned and passed.

1.3 Organizational structure

Third, the organizational structure will be reconsidered by a WG appointed by Nelson Beebe, where the size of the board, the representation (by election, by appointment), etc. will be addressed. With an International TUG in mind, *perceived as such by the people outside the United States !* As an *aside*, some confusion of tasks has been cleared up: the executive director (ED) is no longer a board member! The renewal of the ED-contract can't be done without consent of the board! So TUG will endeavour hard times, and it is expected that the community will experience less service the coming year.

1.4 E-mail voting of BoD

The board of directors (BoD) also motioned the possibility of voting by e-mail, in order to continue work, instead of awaiting the next TUG meeting with an overcrowded BoD agenda. After settling details the motion was passed.

1.5 Reports

Apart from the Language/Local User Groups (LUG s) reports, the following committees reported:

- on local working groups (in writing),
- on membership issues (in writing),
- bylaws (in writing and acting),
- on local working groups (in writing),
- multilingual review (in writing and for review by the membership at large),
- scholarship,
- PD software and its distribution,
- elections/voting by (e-)mail,
- T-shirts (a winner in time spend-revenue ratio),
- finance committee (yes, we will have a loss).

Some 18 committees were listed, with a few missing, especially the one devoted to education.

Much information exchange took place and creative ideas filled the room.

From the written reports the main ideas are summarized below with some oversights added.

1.5.1 LUG s

TUG feels that LUG s are extremely important because they contribute to have TEX etc. widely accepted. We have some well-known and prospering LUG s from Western Europe and Japan, with those from Eastern Europe just taking off. From within the USA the California and Delaware local groups made themselves known.
How to assist?

- Give the welcome feeling (Is formalization of LUG s necessary?).
- Provide information packets:
  for members a resource directory,
  for novices a welcome package (with among others membership/resource directory),
  and for interessees a demo package.
- Provide a list of ideas where LUG boards might pick from in order to serve their members
  How to:
  handle local queries,
  organize resources,
  compile the good information,
  disseminate information,
  distribute books/software/gadgets etc.,
  to organize local meetings,
  to set-up a newsletter,
  start cooperation, ...
- Support local meetings and conferences.
- Support organizing classes, by providing courseware, (names of) teachers, and some funding if needed.
- Share know-how (information/speakers/teachers exchange).
- Fund projects where everybody profits from (TUGboat, LATEX project, TEXHaX, floppy copying machine (passed)).

In return?

- Submission of status reports, participation in board/committees, etc.
- Submission of articles,
- Provide speakers/teachers/authors etc.

Still dangling are:

- The reciprocal/mutual/associate/. . .-membership issues: TUG with any LUG.
- LUG cooperation; an European newsletter/journal in various representations? shared projects? fileservers synchronization?

1.5.2 Membership committee

Axiomatix is expansion of membership. Some proposals made:

- A membership satisfaction survey has been set up and sent out.
- A membership card is urged for.
- Members should be able to obtain discounts and/or special offers on the sale of TEX/TUG materials.
- What about: 'Welcome to membership' packet from TUG and/or LUG?
- What about: 'Information packet with demo floppies or even PD version for free?'
- What about: 'TUG a member?'
- Reciprocal membership TUG with LUG s should be costed out, a pilot study?
- Encourage vendors to help: ask for including as part of a TEX etc. book the TUG/LUG info leaflet; similar enclosure for software sold.

- What about: 'Checking membership list with customer lists, TEXHaX/MaG lists, and inviting non-members to membership?'
- What about: 'Posters with the benefits of TUG/LUG?'
- We should follow-up lapsed members.

Added: Courseware for LUG/TUG beginning course should comprise the information packet.

1.5.3 Working Groups committee

One of the complaints is that WG s don't report regularly. Although this demonstrates that in a volunteer-based organization 'the flesh is stronger than the will,' the board adopted a motion about providing proper guidance to WG s.

1.5.4 Font character encoding committee

Michael Ferguson pushed ahead the difficult standardization process. LUG s are strongly invited to comment on the WG-proposal, sealed by the board.

1.5.5 Fileservers/TUGlib work

Don Hosek and Nelson Beebe have been quite active in this area, not to forget the volunteers out there who feed and maintain the servers. Synchronization of the various fileservers is an ideal task which will never be attained. The long-term goal of course is, to have your local fileserver intelligent enough to handle appropriately your requests. The Achilles heel in any system is to keep the directory up-to-date. Synchronization (and maintaining) tables of contents to strive for is already neat, and useful anyhow. NETlib experience has it that the software used to be available while the directory lagged behind.

1.5.6 (PD) Software validation and distribution

Thinking about software distribution by TUG, especially for personal computers (PC s), has just started. Some LUG s distribute already, apart from the difficult question of a quality/TEX seal (passed Trip/Trap test appropriately, sufficient documentation is provided, etc.) This is different from the former implementers job, because of scale.

A floppy copying machine will be purchased by the office; LUG s can make use of it, at some unknown conditions of yet.

1.6 Home work

During the conference two notes were composed:

- Expanding the use of TEX along with implementation plan, and
- Objectives for TUG 1991–92.

Although these notes reflect most of what was brought to attention (before and at the meetings) they have not been discussed in open and have not been passed, because, . . . , believe it or not, lack of time, other important business, and *no time-control on the various items* of the agenda. For a complete report await the minutes which will be out very soon, because of e-mail acceptance procedure. For the impatients: look out for the president's view of the meetings in next TUGboat or newsletter.

1.7 Next TUG conferences

The very next one is scheduled in summer at the 'home' of TUG: Providence Rhode Island. For the next 'TUG in Europe' meeting offers have been received from GUTenberg and CSTEX. The decision will be made in October.[1]

2 Euro-Summit

For this summit Western and Eastern European 'LUG s' were invited. Present were: GUTenberg, DANTE, ukTEXug, NordicTUG, NTG; Irina from Russia, two delegates from Poland, Yugoslavia, HunTUG, and from CSTUG. For TUG Nelson Beebe, Alan Hoenig and the executive director Ray Goucher, were present; the meeting was chaired by Malcolm, the European coordinator.

After a welcome etc. we started by making ourselves known followed by overviews of the status and wishes of the various LUG s. These surveys as well as the contact addresses were considered useful and therefore Malcolm volunteered in bundling this information (To appear)[2] Moreover, we agreed to extend the Western European information exchange process to the whole of Europe: every LUG will receive newsletters/minutes etc. from the other LUG s. Of course the European coordinator will be informed as well.

The atmosphere of the meeting was cordially, but a bit out of balance to my taste: some 'patronizing' attitude of the Western groups towards the Eastern. Undoubtedly it must have been read as eagerness to assist the 'new' ones. Agreed most Eastern LUG s endeavour difficulties because of *currency conversion* problems, *insufficient network facilities* and in general *hardware drawbacks.*

The Eastern representatives felt it extremely useful just to be present, to participate at the conference, to meet so

[1] GUTenberg pushed ahead. They announced to have the meeting in France, organized by GUTenberg next September, with all the LUG presidents on the program committee, independent from TUG.

[2] This report will partially overlap with the publications:
Summary of resources available to TEX users, TUGboat, 11#1, 32–35, # 2, 207.
The international reports in the proceedings issue, TUGboat, 11#3, 444–450.
TEX, TUG, and Eastern Europe, TUGboat, 11#1, 122–123.

many people with similar interests, to share experience, and to take home most of the Aston archive.

2.1 A summary of characteristics

- **DANTE**
  Most rapidly growing LUG with ≈800 members, a newsletter, two fileservers (also FTP access), listserver, projects: TEX at highschools, TEX on TV, TEX in journals, and several active famous people: Frank and Rainer, Appel, Schwarz, von Bechtolsheim, Wonneberger, Brüggeman-Klein, the LATEX-project guarded by Frank and Rainer.
- **GUTenberg**
  A solid group with ≈400 members, a journal, a listserver, joined fileserver with DANTE, well-attended open meetings, with o.a. TUG/LUG representatives. Offered to host next European meeting.
- **ukTEXug**
  They have Malcolm! (and therefore TEXline, the Exeter proceedings out, firm contacts with other LUG s, . . .), the Ashton archive activity, moderated listserver, several teachers, a TEX companion book in proof, a hundred members or so.
- **NordicTUG**
  Humming TEX and working on character encoding schemes, institutionalisation is fled from like the plague.
- **NTG**
  Roughly a hundred members, two fileservers, a listserver, LATEX activity, bother about teaching and cooperation (especially with other LUG s and SGML), no newsletter yet: just minutes plus appendices (≈ 100p), various contributions to TUGboat.

On the whole Eastern European groups encounter language problems:

- lack of hyphenation tables,
- problems with accented characters,
- problems with (cyrillic)fonts,
- translation of *the* books into the local languages.

They also have problems in just getting the materials there. Most groups are not yet formally organized. At the moment it is unclear in what way help could be given apart from exchanging contact adresses, information, software and the news.[3] Note that Bien and Ryśko already published articles in TUGboat.

3 Birds-Of-a-Feather sessions

Of the several BoF s one did address a motion to the board.

3.1 Future of TEX

The board is asked to acknowledge that TUG and the LUG s have to maintain and develop TEX etc. That it should oversee and coordinate changes to TEX in order to prevent fragmentation, that it shall find a balance between stifling and development, that it shall stimulate and/or fund research into unsolved typographical problems! This motion has not yet been discussed by the board, because, . . . , yes, you know already.[4]

4 Conference

A separate conference report will appear written by Nico, Johannes and myself.

[3] More detailed information: see the earlier announced Euro-Summit90 report

[4] Added in proof: See Knuth's statement enclosed as appendix.

BIJLAGE Y

The Dutch and international 1990 TEX conferences

N.A.F.M. Poppelier, C.G. van der Laan

10–13 September 1990

The fifth European TEX conference was also the first TEX Users Group Meeting in Europe, i.e. the first international meeting of the users group outside North-America. The conference was held at University College Cork, a university campus with a charming mix of old and new buildings among lawns and trees, in Ireland's second largest city.

The organizer of the conference, Peter Flynn, who works at the computer centre of University College Cork, had done a great job on organizing this big conference with about 175 attendees from 23 (!) different countries, with the notable exception of Japan. As for the social side of the conference: there was a get-together party after registration on Sunday evening, a dinner at the university restaurant on Monday, and on Tuesday an excursion to Blarney Castle, followed by dinner with live music and folk dancing in a restaurant nearby. Apart from this there was also plenty of opportunity to meet people and discuss various things during coffee, lunch or tea breaks. The programme was not overfull and there was more than enough time to discuss TEX and related matters with TEX users from all over the world.

Monday, September 10

After not-so-formal opening speeches by the Minister for Science & Technology of Ireland and the dean of University College Cork, the first international TEX conference in Europe was a fact.

Erich Neuwirth (University of Vienna) had the honour (unpleasant task?) of giving the first talk. He showed that you can create quick-and-diry databases with basic tools such as TEX and the Unix tool `awk`. Erich seemed to advertise the use of `awk` by larger parts of the TEX community: `awk` has been ported to the MS-DOS world and, apart from several commercial versions, there is a free version available from the Free Software Foundation.

In her talk on TEX and hypertext, *Christine Detig* (University of Darmstadt) gave several definitions of hypertext, and discussed advantages and disadvantages of the hypertext approach. The conclusion of her talk was that the future of electronic publishing *could* be an interwoven electronic process with the following elements: (1) hypertext and logical markup, (2) TEX for typesetting, (3) drivers for various screen and printer types. One criticism: discussions about 'hypertext' can only be useful if all people attending the talk have *seen* hypertext – describing it on paper or transparencies is certainly not enough to get an idea of what it is, or what it can be.

Les Carr (University of Southampton): 'Experiments with TEX and hyperactivity'. Les Carr reported on work, done by Sebastian Rahtz and him, based on the question: How can existing (TEX) marked up copy be used in a hypertext system? Particularly amusing was the division of users/developers of these kinds of systems into the 'fluffies' and the 'technoids'.

Problems tackled include the use of TEX to format text being displayed on screen in variable sized windows, embedding generic hypertext navigation tools in the source and using a single source to generate both printed and hypertext versions of a document.

The authors concluded with: 'It is clear that TEX in a hyperactive world can only survive in a generic markup form such as LATEX. Much of its power (such as pagination) is irrelevant in this context, and its implementation language will continue to put off many potential designers. But TEX retains a raw beauty of its own, and if we decide that the hyper systems we build must have a formatting engine behind them, we are confident that TEX will continue to be the first choice for many years to come.'

The talk by *Johannes Braams* (PTT Neher Laboratory), 'The Dutch N ational LATEX effort" was a report on the work done by working group 13 of the Dutch TEX Users Group. He discussed the development of a set of document styles that are compatible with the standard document styles of LATEX, but have a layout that appeals more to Western-European tastes, and the work on 'internationalizing' LATEX, i.e. adding tools to LATEX that allow an author to switch between different languages, a project that bears the appropriate name 'Babel'.

Adrian Clark (Essex University): 'Documenting a TEX archive'. The key issue addressed was: 'An archive

This interim report will finally be published in Malcolm Clark's favourite magazine: TEXline.

is only as good as its documentation.' Where is the archive? How to locate the software? How to get it across? The emphasis was put on the second question. Experience has it that *functionality* requests precede name requests. Keywords and help systems are the tools, along with the good old systematic index, provided there is a generally accepted one. The author reported about a proto system for automatically maintaining the 'index' of the archive, the help system etc. In order to have this kind of systems working, contributors must be persuaded to obey the submission rules: along with the software, documentation in a certain standard format must be supplied. Perhaps overlooked is the issue of a general accepted classification index. Even within the field of numerical software several indices are around.

During the lively discussion after the presentation, useful ideas about what users would like to see were proposed, ranging from:

- the ideal situation, just your own local archive 'at the corner' with *virtually* everything,
- to documentation requirements (time stamps, version indication, minimal tests, quality, refereeing, etc.).

Nelson Beebe mentioned the synchronisation effort of the various archives and the TUGlib project. These projects have the same kind of problems.

Tuesday, September 11

Thomas Kneser (GWD Göttingen), the first speaker on Tuesday morning, presented an adaptation of the work of Thomas Reid to a LATEX context. Thomas Reid has developed macros that allow to have paragraphs of text to 'wrap around' figures. Thomas Kneser showed that these macros can be successfully adapted to the LATEX case. However, in some cases a certain amount of cheating is necessary to get the desired result.

In his talk, 'The document style designer as separate entity', *Victor Eijkhout* (TEXTechniek) discussed the need for a separation of all tasks in a TEX-based system in three 'layers': (i) the author, who uses a document style prepared for a certain class of documents, (ii) the document-style designer, who creates a document style from a TEX-based toolbox, and (iii) the TEX expert who creates and maintains the toolbox. He argued the need for a programmable toolbox, so that the style designer could write TEX macros without programming in TEX; this point was illustrated with examples from a format the speaker developed.

'QuickDraw, PostScript, TEX' by *Tim Murphy* (Trinity College Dublin) was a rather confusing talk. Afterwards I was unable to summarize what it was he had tried to tell us. A lot, I know, but it lacked coherence. What I did pick up was this: he loves GRIF, but hates SGML, for reasons that are still unknown to me. He appeared to be in favour of the work, done by a Euromath committee, on defining a complete syntax of mathematical formulae for GRIF. He advocated the basic idea of GRIF of letting a document be generated by a context-free grammar.

M. Maclenan (South Bank Polytechnic) considered himself to be very new to TEX and definitely not an expert, but he showed some very interesting TEXnical stuff. At the South Bank Polytechnic there arose the need for a tool for drawing circuit diagrams for text books. PICTEX, although it requires a lot of memory and is rather slow, can produce impressive full-page circuit diagrams. The speaker showed that a form of object-oriented programming can help a lot. In the discussion after the presentation, speaker and audience came to the conclusion that an interesting extension – and a possible time-saver – would be a collection of symbols and parts of diagrams created with METAFONT. Another future development could be integration of TEX with a CAD package: extract a list of (x, y) coordinates from the CAD programme and use this as input to a TEX macro package.

The talk by *Rainer Schöpf* (University of Heidelberg), 'Towards LATEX 3.0', was a list of ideas for the new version of LATEX. He outlined some ideas for a future version of LaTeX, but cautioned that none of these ideas were definite yet. Much importance will be given to the style designer interface.

For about a year now, Frank Mittelbach and Rainer Schöpf, have been discussing ideas – mostly via electronic mail – for a new version of LATEX with a group of TEX, LATEX and document-style experts in Europe and the United States. A proposal for the new design and a few prototype parts of the new package are expected shortly.

The presentation of *Brian Hamilton Kelly* (Cranfield Institute of Technology) was the last one before the visit to Blarney castle in the afternoon. His presentation focussed on a new public domain implementation of TEX 3.0 and METAFONT 2.0 for VAX/VMS. The distribution contains normal and 'big' versions of TEX, LATEX and SliTEX. Useful facilities: TEX's `e` option really works, TEX, LATEX and SliTEX now available as VMS command verbs. It has recently been extended by Don Hosek and is now also the DECUS implementation.

Wednesday, September 12

Malcolm Clark (Imperial Cancer Research Fund Labs), was kindly asked to change the title of his presentation, 'Post Congress Tristesse', even though most attendees didn't understand why. His presentation was a very enlightening one for those TEXies interested in doing book projects with TEX. The topic of his talk was the process of making the proceedings of the TEX'88 conference at Exeter ready for publication. To discourage anyone with ideas of being an editor in the near future, here's a list of the problems Malcolm described:

- variations in medium (disk, electronic mail and even paper)
- character coding (ASCII, EBCDIC)

- late submission (please wait—wait some more—nag— threaten to print just the abstract)
- input form (plain TEX, LATEX). In the end, everything was converted to plain TEX with additional macros to reproduce a layout similar to the layout of Addison-Wesley's computer science series.
- a professional indexer was hired to compile the index afterwards: the result was bad.

An important decision: do you referee papers or include all of them? The speaker's view was that you should not edit or referee a paper: if someone wants to present work with TEX or METAFONT, he or she should be free to do so. Removing syntactical errors is acceptable, any further editing of the English – or almost English – isn't.

A matter of style: who determines the layout, the editor or the author? Answer: the editor, so the authors should not markup (too much). In some cases plain ASCII is better than **TEX.

Finding a publisher: Addison-Wesley (not interested), Springer (no contacts with them), John Wiley & Sons (never publish proceedings). Finally Ellis-Horwood published the proceedings. Advantages of E-H: no control, you can do whatever you like as editor. Disadvantages: you have to do everything, upto and including producing the bromide!

Conclusions:
- Publishers like it when authors do all the work *for them*, which is a really funny view!
- Publishers should not accept 300 dpi camera-ready copy, since this is without exception of appalling quality.
- Computer Modern is an excellent font at 1270 dpi, but looks horrifying at 300 dpi. Books produced from low-resolution output are no advertisement for the high quality of TEX's typesetting.
- Adobe and other font producers still have to come up with a nice font that includes math!
- Copy editors are essential. However, some publishers accept manuscripts in electronic form, or even offer the possibility of electronic mail submission, and then eliminate technical editing from the publication process.
- Working with other amateurs on any project means sooner or later that they withdraw as soon as other things become more important.
- Expect no thanks.
- It's fun!

'TEX & SGML' by *Kees van der Laan* (University of Groningen) was a talk that could have been more interesting if more people in the audience would have had some experience with SGML. It is not enough to give the following description: 'SGML is a meta-language that is a means to express your thoughts on the entire lifecycle of documents', that serves as a standard for tagging and interchanging documents, and that provides a mechanism for representing special characters, tables, and mathematical and chemical formulae, using ASCII coding. The speaker has recently investigated the problems with conversion of material coded in TEX or LATEX to SGML, especially mathematical formulae and tables.

If SGML becomes a household word in electronic publishing and publishers provide their authors with *document type definitions* (dtd's), authors have to be aware at all times of this dtd and the tags, entities and attributes it defines. The publishers need standardized general-purpose dtd's or SGML experts who develop new dtd's.

Another problem of using SGML for the coding of mathematical formulae and tables is the apparent impossibility of separating form from content.

Publishers and other users of SGML often state that one of the advantages of SGML is that it allows one to store information such that it can be used for other purposes later on. One of the conclusions of the speaker was that re-usage is maybe not that important since scientists tend to continually rewrite and update their articles and books. Another conclusion was that the meta-ness of SGML is a strong point of SGML, but at the same time also one of its weakest points.

Olivier Nicole (INRA) presented another use of the PICTEX package. He used it to printing graphs that were generated by the statistical package S. From both talks on PICTEX at this conference it became clear that it is a very useful and powerful package: the PICTEX macro package can be found in several electronic archives, and there is a very good manual available. However, it is very slow, you need a big version of TEX and a 10 kb `tex` file can result in a `dvi` file that is ten times as large!

The rest of the day was devoted to font design with METAFONT and various applications of such fonts.

Alan Hoenig (CUNY) presented a solution to the problem of labelling figures in TEX documents. His message was: graphics are no longer a problem for TEX (?), but the problem is to get labels in the figures that use the same font as the text. The basic strategy is the following: create a picture with METAFONT, write the coordinates of points to be labelled as `fontdimen`'s to the `tfm` file. In the TEX document you pick up the x and y coordinates from the `\fontdimen`'s.

METAFONT lacks read and write operations, so (ab)using the `tfm` file for this purpose is the easiest solution. Furthermore, writing a parser in TEX to decode information in the log file of a METAFONT run would be, to say the least, cumbersome.

Final remark: to make it easier to use METAFONT for creating all sorts of pictures, one needs a macro package 'on top' of it, similar to LATEX 'on top of' TEX.

'Typesetting Old German' by *Yannis Haralambous* (Université de Lille) was one of the nicest talks of the entire conference. It was awarded the prize for best presentation at this conference. The speaker started out

by observing that many people believe that old (early) music can be best enjoyed when played on old instruments. If you draw a parallel with reading old texts: these can maybe be enjoyed more when they have been printed with old typefaces. The speaker showed pages from the Gutenberg 42-line bible and a book by C.Ph.E. Bach. To re-create these texts, the speaker has designed METAFONT fonts for Gotisch, Schwabacher, Fraktur and ornamented capital letters – the ornamented capitals where beautiful and extremely well done, enough for a big round of applause. While working on the Fraktur it turned out that the old designers applied Bezier curves without knowing it: a lot of the curves can be specified by only a few pen positions.

As an aside the speaker has also created an Arabic font, of which he also showed examples. His next project will probably Renaissance Greek, where you need a font with ± 400 ligatures.

Mícheál Ó Searcóid began his presentation 'Irish letter forms with METAFONT' with a short history of writing in Ireland. Elizabeth I ordered the design of an Irish typeface, modelled after written letter forms. Use of this typeface started around 1571. Through history there have been several re-designs. After the historical introduction, the speaker showed an example of modern Irish letter forms, created with METAFONT.

The next presentation, on 'An international phonetic alphabet', was by *Dean Guenther* (Washington State University) and Janene Winter. Dean is manager of the project and Janene was the METAFONT specialist. The project was started because of frequent requests from the humanities departments. At first, some papers were published with Computer Modern fonts and handwritten phonetic symbols. Later on, phonetic symbols were created as bitmaps. However, these only looked good with Almost Modern Roman, so a new font was needed when people started using CM. There were lots of obstacles: MF-84 came when Janene had just learned MF-79, there was no help on campus, there was no previewer. But the result is an IPA font, to be precise: a 128-symbol subset of one of the varieties of IPA.

Since Janene now works for the AMS, the work on the IPA project has unfortunately stopped. At the State University of Washington a different IPA project has started, partially based on the work done at WSU.

Adrian Clark (Essex University): 'Halftone output from TEX'.

Thursday, September 13

Nico Poppelier (Elsevier Science Publishers): 'SGML & TEX in scientific publishing'. This presentation was non-technical. The speaker's idea was not to discuss the 'What and how?', but the 'What and why?'. What can SGML mean to a publisher, i.e. what role can it play? SGML and TEX can be be a nice combination, but so can combinations of SGML with other sophisticated text-processing systems and/or typesetting systems.

The speaker also discussed

- some of the advantages of LATEX over plain TEX, both for journal and for book publishing, and
- other uses of TEX within Elsevier Science Publishers, namely as the back-end of a database-publishing system

The presentation by *Amy Henrickson* (TEXnology), 'Getting TEXnical' was a 30-minute crash course in TEX macro writing. A series of useful techniques, lots of examples, a few basic and a few not-so-basic TEX tricks. Interesting talk, but with a very high pace.

Frank Mittelbach (EDS, Rüsselsheim) intended to give a talk 'New BibTEX requirements' instead of having an open discussion with this title, as was announced in the program. The main points of Frank were that

- BIBTEX is not suitable for applications in the humanities or, e.g., for books with bibliographies in every chapter
- BIBTEX does not fit well with LATEX 3.0
- adapting BIBTEX to other languages than English is not always possible

He proposed several major changes to BIBTEX; the complete list is too big to be repeated here, but the paper will appear in the proceedings.

The title of the presentation by *Angela Barden* (City of Cork Vocational Education Committee) was intriguing: 'Purchasing pain with all that joy'. It turned out to be an interesting, and sometimes amusing, examination of several books on TEX. The speaker, who is a teacher of reading skills, examined the following books:

- 'The TEX book' by Don Knuth. It tells lies and jokes (even in the index, where it is out of place), hides information in exercises, contains exercises that are too difficult, the often confusing dangerous bends, and conflicting instructions. On the whole, a user-unfriendly book.
- 'The Joy of TEX' by Michael Spivak. The first exercise in the book was pointless, so why do the others? And the answer given in the back was wrong and unintelligible. The term 'macro' is never defined. 'Maybe it helps mathematicians, but not me', was the speaker's conclusion.
- The PC-TEX manual, by the same author. A page-numbering scheme and a confusing structure. For example: there is an appendix A and an appendix F, but no other appendices. The speaker abandoned the book at an early stage.
- 'LATEX user's guide and reference manual', by Leslie Lamport. The speaker was grateful to the author for his straightforward way of presenting information. No tedious exercises, a no-nonsense way of writing, clear concise prose.
- 'TEX for the impatient', by Abrahams, Hargreaves and Berry. Not radically different from the other books. A dull layout, an enormous amount of print

on a page, turgid style, but not as bad as the other books.
- 'A gentle introduction to TEX', by Michael Doob. Not a bad book, but unfortunately there are exercises.

Advice for future books: tell no jokes, be careful about the imagery you use (if TEX has eyes, mouth and stomach, what is the analogon of `\shipout`?), use shading or another technique to distinguish input from output, use illustrations to clarify concepts, separate text book from manual, give more examples, don't hide information in exercises.

In his presentation 'TEX in schools, just say no', *Konrad Neuwirth* argued against teaching TEX in schools. There doesn't seem be a class where TEX would be appropriate: it doesn't teach you anything about math, writing math or physics reports with TEX in class takes more time than the usual 50–60 minutes, and as an instrument in teaching aesthetics TEX is also questionable, since most TEX documents look horrible to a trained eye. Furthermore: TEX is not the most user-friendly of programs.

Philip Taylor's talk 'Improving the æsthetics of mixed font documents' was an account of the work on the book 'Principles of nutritional assessment', which was recently published by Oxford University Press. According to the speaker, Don Knuth's final exhortation in the TEX book, 'Go forth and create masterpieces of the publishing art', has developed into 'Go forth and create horribly looking books that shriek TEX(LATEX)!'

Letting TEX typeset a book with a pleasing design, using Postscript fonts (Times Roman) for text and Computer Modern for math, and with lots of tables and illustrations turned out to be a far from trivial task. The resulting book was passed around during the presentation and proved to be an attractive book, impeccably reproducing the sort of looks one expects from the Oxford University Press, where it was published. One small criticism: Philip had forgotten to turn on `\frenchspacing`.

After a rather long introduction on the history of various page patterns and the document types in which they occur, the next speaker, *Alan Wittbecker* (DEC), came down to business: 'ArchiTEX as an international page pattern maker'. ArchiTEX is a macro package that he developed to re-create the variety of page patterns he described, using TEX. For that, a rectangular grid is put 'on top of the page'. To every intersection point of the grid a box of text can be attached.

The last presentation of the conference was 'Integration of graphics into TEX', by *Friedhelm Sowa*. The idea behind the speaker's approach is to convert the output of graphics packages, in this case files in `tif` format, to TEX's `pk` files. A picture is divided into tiles of certain dimensions, and the tiles are converted to characters in a TEX-type font. TEX is then used to glue the tiles together to produce the original picture. The speaker discussed various methods of obtaining half-tone pictures, with varying number of grey shades and with or without error distribution, and showed several examples.

Birds-of-a-feather sessions

The organisers of a TEX conference had a golden moment when they came up with the idea of bof. The idea is simple: on every day of the conference programme the organization allocate one or more time slots of, say, half an hour. Anyone with ideas for a discussion can allocate a time slot and . . . just go ahead! In other words: the programme provides time especially for informal or improvised discussions.

At the Cork conference, there were birds-of-a-feather sessions, or bof's, on graphics, the future of TEX, and LATEX 3.0. The bof on the future of TEX was especially interesting, since it became a three-hour discussion that resulted in a petition to the board of TUG and Don Knuth, signed by some 25–30 concerned TUG members.

One of the major problems at present is the fact that the creators of TEX, LATEX, BIBTEXand MakeIndex have, for various reasons, decided to stop working on their programs. In some cases they have announced that they consider their programs as mature, i.e. will not develop them any further. This is a matter of great concern to many TUG members and to the board of TUG as well.

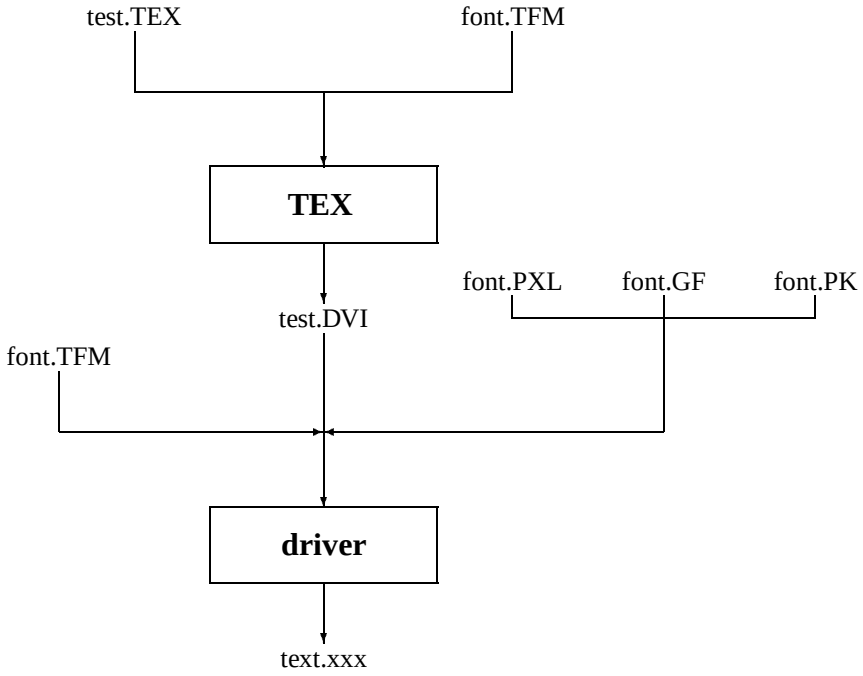
BIJLAGE Z

TEX structuurschema's

Huub Mulders

1 De structuur van TEX

Hieronder volgt een schema waarin is aangegeven welke bestanden er worden gebruikt en aangemaakt wanneer met behulp van TEX van een tekstbestand een *gezette* tekst wordt gemaakt.



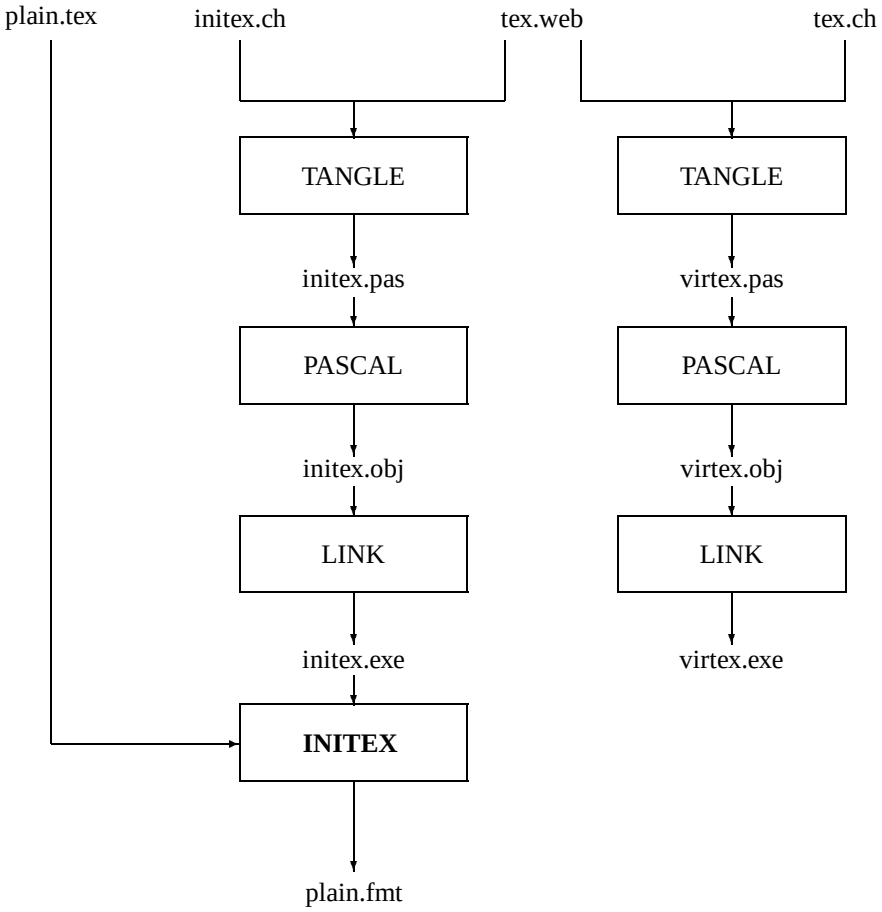
2 Overzicht filetypes die bij TEX etc. een rol spelen

Hieronder volgt een overzicht van de verschillende filetypes, aangegeven door een *suffix*, die bij TEX een rol spelen. Merk op dat sommige besturingssystemen, zoals MS/PC-DOS, de *suffixes* beperken tot drie symbolen.

Specifiek voor TEX etc.		
.aux	:	TEX hulpfile; gemaakt en gebruikt door TEX (en o.a. LATEX)
.base	:	METAFONT format-file (geprecompileerde macro's, "plain"); gemaakt door INIMF, gebruikt door MF of gepreload door VIRMF
.bbl	:	BIBTEX outputfile; gebruikt door LATEX
.bib	:	BIBTEX inputfile
.bst	:	BIBTEX documentstyle
.ch	:	aanpassingen van .web-files; gebruikt door TANGLE en WEAVE
.doc	:	1. documentatie
	:	2. LATEX documentstyle, met kommentaar (verder gelijk aan .sty)
.dvi	:	device independant document; gemaakt door TEX; gebruikt door drivers
.fmt	:	TEX format-file (geprecompileerde macro's, "plain") gemaakt door INITEX, gebruikt door TEX of gepreload door VIRTEX
.*gf	:	generic font: pixels (*=300xvergroting); gemaakt door METAFONT, gebruikt door drivers
.glo	:	LATEX glossary (hulpfile)
.idx	:	LATEX index (hulpfile)
.lis	:	o.a. TEX/METAFONT logfile
.lof	:	LATEX list-of-figures (hulpfile)
.lot	:	LATEX list-of-tables (hulpfile)
.mf	:	METAFONT inputfile
.*pk	:	packed font pixels (*=300xvergroting); gemaakt uit .*gf door GFTOPK, gebruikt door drivers
.pool	:	stringpool; gemaakt door TANGLE, gebruikt door bijbehorende .exe
.*pxl	:	font pixels (*=1500xvergroting); gemaakt uit .*gf door GFTOPXL, gebruikt door drivers
.sty	:	LATEX documentstyle of style-optie
.tex	:	TEX/LATEX/SLITEX inputfile
.toc	:	TEX table-of-contents (hulpfile)
.typ	:	"leesbaar" gemaakte .dvi; gemaakt door DVITYPE
.tfm	:	TEX font metrics; gemaakt door METAFONT, gebruikt door TEX drivers
.web	:	sources van diverse programma's; gebruikt door TANGLE en WEAVE

3 Het genereren van een nieuwe TEX versie

Hieronder volgt een schema waarin staat aangegeven hoe een nieuwe versie van de TEX programmatuur kan worden gemaakt. Hier wordt uitgegaan van PASCAL als programmeertaal. De mogelijkheid bestaat ook om in een andere programmeertaal te werken, zoals C. In dat geval is er een speciale versie van WEB nodig.



Het maken van een `TeX.exe` en `LaTeX.exe` is o.a. afhankelijk van de gebruikte TeX.ch en/of van welke 'firma' de TEX vandaan komt.

- Voor de TeX van de DECUS-tape:

- Voor Kellerman & Smith moeten bij het linken van initex.obj en virtex.obj een aantal svi*.obj files mee-gelinkt worden. Dan kan `(La)TeX.exe` gemaakt worden zoals bij DECUS-tape.
- Meestal is er wel een documentatie file waarin staat hoe een `TeX.exe` te maken
- TEX en LATEX kunnen altijd gedraaid worden via het commando:

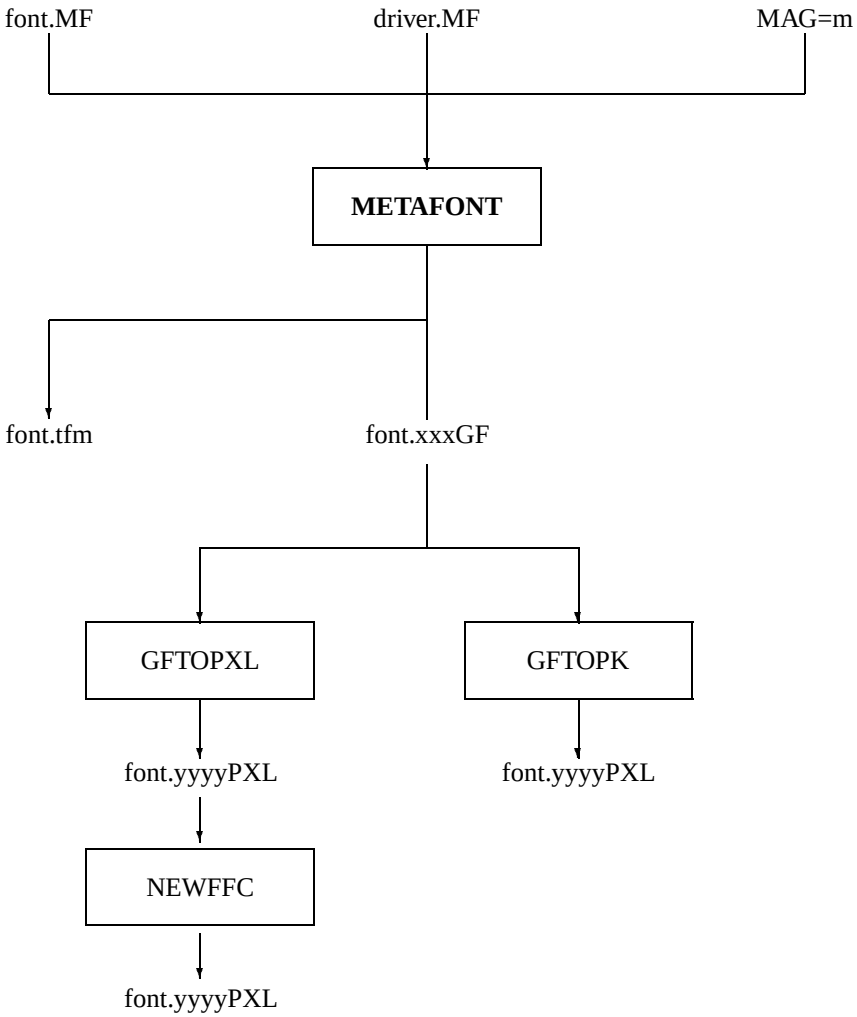
```
VIRTEX &plain filename
```

of:

```
VIRTEX &lplain filename
```

4 De structuur van METAFONT

Hieronder staat schematisch aangegeven welke bestanden er worden gebruikt en aangemaakt wanneer met METAFONT *fonts* worden gemaakt. In het schema zijn ook diverse nabehandelingsprogramma's (`..TO..`) aangegeven:



Vergrotingen en naamgeving van fontfiles

magstep		0	0.5	1	2	3	4	5	6	7	8	9
mag*1000	(m)	1000	1095	1200	1440	1728	2074	2488	2986	3583	4300	5160
gf	(xxx)	300	329	360	432	518	622	746	896	1075	1290	1548
pxl	(yyyy)	1500	1643	1800	2160	2592	3110	3732	4479	5375	6450	7740

BIJLAGE AA

The 1990 DECUS TEX Collection

Announcement
Ted Nieland

(news from TeXHaX issue 62)
4 september 1990

The DECUS Languages and Tools SIG Public Domain Working Group and the Electronic Publishing SIG TEX/LATEX/WEB Working Group are proud to announce the 1990 DECUS TEX Collection. This collection offers nearly everything a TEX User would want on their system for TEX.

The master tapes for the collection have been sent to DECUS Library and to the top of the DECUS LUG distribution tree. The new collection will be available to all shortly through their channel for procuring DECUS Software.

This collection is an extensive rework of the previous collection with nearly all of the material being updated or new. More DVI drivers have been added and many of the VMS programs now sport a CLD interface.

Also, an extensive effort on documentation has taken place resulting in a DECUS TEX Help Library.

The following items are included in the DECUS TEX Collection 1990:

- WEB (Tangle 4.0 / Weave 4.1)
- TEX Version 3.0
- METAFONT Version 2.0
- LATEX Macro Package 2.09 <7 Dec 1989> (with mod for TEX 3.0)
- SliTEX Macro Package 2.09 <7 Dec 1989> (with mod for TEX 3.0)
- BibTEX Version 0.99c
- TEXsis Macro Package Version 2.13
- DVIOUT Version 1.2
- DVIPS for VMS, Version 5.35
- DVItoVDU Version 3.2
- DVItoLN03 Version 3.1-4
- XDVI (with support for DecWindows)
- TEXx (with support for DecWindows)
- Vassar Spell Version 2.2
- FWEB (including support for VMs)
- CWEB (including support for VMS)
- MWEB
- TIB
- CRUDETYPE
- DVIDIS (for VAXStations Running VWS)
- GPLOT 4.23
- RNOTOTEX
- IDXTEX
- GloTEX
- DVIDVI
- MAKEINDEX
- PicTeX
- TEXTYL
- DVI2TTY
- LSE Templates for LATEX and BibTEX
- MFWARE (GFtoPK, GFtoPX, etc)
- PICMODE
- TR2TEX
- WS2LATEX
- AmS-TEX Macro Package
- AmS-LATEX Macro Package
- PHYZXX Macro Package
- PHYSE Macro Package
- ScriptTeX Macro Package
- MuTEX Package (including METAFONT files)
- Clarkson LATEX & BibTEX Style Collections
- DECUS TeX Help Library
- Beebe Utah DVI Driver Collection with additional submissions
- DVI2PS
- Many Font Additions (Concrete, Duerer, Chess, DECUSLOGO, among others)
- Support for Foreign Languages including Dutch, French, German, Greek, Hebrew, Icelandic, Italian, Japanese, Korean, Portuguese, Russian, Spanish, Thai, Turkish, and Vietnamese.
- TEX for the Amiga with some DVI Drivers and the LATEX Picture Editor (LPE)
- TEX for the Macintosh (OzTEX), along with BibTEX, and DVI drivers.
- TEX for MS-DOS, plus previewers and DVI drivers.
- Various TEXware for UNIX, including WEB2C and XTEX (for DECStations).

The following output devices are supported:

- DEC LN03 (requires a RAM Cartridge) [DVITOLN03]
- DEC LN03 Plus (uses bitmaps) [DVIL3P]
- DEC LA75 [DVI175]
- PostScript (LPS40, Apple LaserWriter, LN03S) [DVIALW, DVIPS, DVIOUT, GTEX]
- Hewlett Packard Laserjet [DVIJET]
- Hewlett Packard Laserjet Plus [DVIJEP]

- Cannon Engine Laserprinter [DVICAN]
- EPSON Printer [DVIEPS]
- Printronix Printer [DVIPRX]
- Okidata Pacemark 2410 (72 or 144 DPI) [DVIOKI]
- VT terminals, ReGIS Terminals, Tektronix Terminals [DVITOVDU]
- VAXStations running VWS [DVIDIS]
- DECWindows [XDVI, TEXX]
- Version 3.10 BBN BitGraph Terminal [DVIBIT]
- Golden Dawn Golden Laser 100 printer [DVIGD]
- Imagen imPRESS-language laser printer family [DVIIMP]
- Apple Imagewriter 72 or 144 dpi printers [DVIM72 or DVIMAC]
- MPI Sprinter 72 dpi printer [DVIMPI]
- Toshiba P-1351 180 dpi printer [DVITOS]
- Generic Output [DVI2TTY]
- QMS Laser Printers [GTEX]

The collection includes numerous example files including A Gentle Introduction to TeX by Micheal Doob and Essential LaTeX by Jon Warbrick.

For more information on getting a copy of the DECUS TeX Collection, contact your DECUS Local User Group or the DECUS Library at:

DECUS Library (BP02)
219 Boston Post Road
Marlboro, MA 01752-1850
(508) 480 3418/3659/3446

BIJLAGE BB

New Books on TEX

Victor Eijkhout
University of Illinois at Urbana-Champaign
305 Talbot Lab
104 S. Wright Street
Urbana, Illinois 61801, USA
u641001@HNYKUN11

There is a piece of good news to be reported: two new books on TEX have appeared recently, one for beginning to intermediate users, and one for intermediate to advanced users. And there's more good news: TEX is so widely spread that both books originated in Germany, and are written in German. Of the introductory book translations into English and Dutch exist, but the advanced book, in more than one respect the more interesting of the two, has not been translated yet.

TEX books in other languages than English are a good thing for two reasons. One is that they give an indication of the widespread use of TEX. The other is that, quote Norbert Schwartz, author of 'Einführung in TEX' [1], such books are 'a bit more internationally oriented than a book of English or American origin would probably be'. This is especially apparent in 'TEXfür Fortgeschrittene' [2] by Wolgang Appelt, which has a whole chapter on 'Deutschsprachige Text', containing sensible remarks that are relevant to more languages than just German – although they are not particularly relevant to the somewhat plebeian English language.

Introduction to TEX

'Introduction to TEX' by Norbert Schwartz assumes no knowledge of TEX whatsoever, indeed the first chapter 'General information' gives a short list of the merits of TEX. This makes for a nice and motivating introduction for the complete novice.

The same holds for chapter 2, 'Operation', that contains, after a few pages of braces, backslashes, and punctuation, a first example of the use of TEX. Some thirty commands are used here. Obviously the author wants to get the reader going: the details will come later.

Chapter 3 was written in the same vein. In 30 pages a large amount of information about 'Setting text' is given to the reader, with lots of examples. However, this chapter had me frowning a number of times. It is the author's style of writing to use bursts of creativity like

```
\obeylines\everypar{\hfil}
```

to introduce the concepts of `\everypar` and `\parfillskip`, but it wouldn't be mine. And I object to

```
\centerline{\it The current page has
    the number \folio}
```

Fortunately, some important concepts are explained more fully in chapters on macros and 'How TEX works' – although I feel that the section on modes is a bit skimpy. There are two nice chapters on mathematical typesetting, there is a short chapter on output routines, and I was particularly pleased with the chapter on 'Tables and alignment'.

Main part of the appendix to this book is an 80 page (!) list of all TEX and plain TEX commands. The explanations are short, but certainly not cryptic, and often an illuminating example is given. Definitely a good idea of the author.

TEX for the advanced

'TEX for the advanced' by Wolfgang Appelt is a very different book. The subtitle, 'Programming techniques and macro packages' is probably the best indication to its contents. Wolfgang Appelt argues in the preface the need for high level macro packages, and then sets out to assist the reader in constructing such packages. He does this in three ways.

The preface, the introductory chapter, and a chapter 'Macro packages' give general thoughts on how macro packages should be structured, and what their nature should be. He distinguishes between the logical structure and the layout structure, and for both of these the generic and the specific structure. It is useful to have such concepts explained in some detail, and the reader won't hear me arguing the author's point of view.

Pure TEX theory is treated on chapters on 'Spaces' – such a chapter must be answering many prayers of desperate TEXers – and 'Macros and parameters'. The author has a very clear style of explaining, but his explanation of conditionals, sufficient for most cases, distorts the truth a bit.

Lastly, four chapters can be classified as 'case studies in macro package design'. They treat the subjects of a font selection scheme, text structures (lists and sectioning), referencing (including table of contents), and adaptations necessary for the German language. These chapters give complete sets of macros, and they are well explained.

Appelt makes no attempt at being complete. Mathematical typesetting and alignments are not treated in this book, and output routines are hardly touched upon. Given the size of the book this would not have been possible, and concentrating on a few selected topics is probably a good idea.

In all, this book is maybe not sufficient reason to start learning German – which means you'll never make such delightful discoveries as that 'ragged right' is 'Flattersatz', 'flutter setting' in German – but if you know a smattering of the language it certainly won't harm you to pick up this book.

How does it look?

When a book about TEX appears, there is an obvious question: 'has it been done in TEX?' For both books reviewed here the answer is yes, but the results are widely different. The Appelt book is set in 12 point Computer Modern with non-obtrusive headings, which gives a surprisingly open and readable page. Of the Schwartz book I have only seen the Dutch and English translations, which are totally unlike each other. The English translation is set in Computer Modern, photographically reduced to 10 point. Unfortunately, the book was printed rather lightly, which makes the page appear somewhat vague.

The Dutch branch of Addison-Wesley must have been in an adventurous mood, combining New Century Schoolbook as a text face with Avant Garde headings. Choosing Courier as the type writer font was not the optimal choice, but the overall result is rather pleasant – even though there have been a few accidents in typesetting the examples.

As a conclusion I would state that both books are an asset to the TEX community. Neither book is a definite TEX bible, but for both there is certainly a niche to fill.

[1] Einführung in TEX, Norbert Schwartz, Addison-Wesley Verlag, Bonn 1988(?) ISBN 3-925118-97-7
Inleiding TEX, Norbert Schwartz, Addison-Wesley Europe, Amsterdam 1990, ISBN 90-6789-151-7
Introduction to TEX, Jost Krieger and Norbert Schwartz, Addison-Wesley Europe, Amsterdam 1989, ISBN 0-201-51141-X
[2] TEX für Fortgeschrittene, Wolfgang Appelt, Addison-Wesley Verlag, Bonn, 1988, ISBN 3-89319-115-1

TEX for the impatient

One of the aspects of TEX that set it apart from other text processors is the fact that there exists an ultimate reference: the TEX book. Like its introduction states, this book is for people who have never used it before and the experienced hackers alike. 'TEX for the impatient' makes a similar claim: Paul Abrahams, the senior author, asked himself 'What kind of book would have made it easier for me to learn TEX? What kind of book would I need now, as a more experienced user, to locate commands or functions that I never learned or only half remember?'. In my opinion he, and co-authors Karl Berry and Kathryn Hargreaves, have given an succesful answer to the first question. My thoughts on the second question follow below.

'TEX for the impatient' has a very appealing front cover: the white rabbit from 'Alice in Wonderland' (the one that exclaims 'Oh dear, I shall be too late!') is sitting, looking at his watch, very impatiently. The inside of the book looks good. Computer Modern is used for the text, with a surprising but very satisfactory choice of Optima bold for headings and command names when these are used as headings. Thirteen chapters and an index make up the approximately 360 pages of the book.

After two inevitable chapters 'Using this book' and 'Using TEX', follows an interesting third chapter: 'Examples'. Ten page-long examples with the input on the facing page give a good impression of TEX's capabilities, and give the novice a source of commands and constructs to study (and copy).

Chapter four 'Concepts' starts the reference part of the book. Instead of merging the list of concepts treated here into the table of contents, the authors decided to print it separately on the inside of the back cover. An unusual idea, but I like it. The list is some 90 terms long, and the chapter spans 55 pages. Individual concepts are therefore treated briefly but the explanations are clear and well-written, and there are many references to the subsequent chapters which treat individual commands. In this chapter I appreciated especially the fact that the authors use the anatomical analogy for TEX's workings, and refer to it repeatedly.

Although the authors suggest that novices, after having read chapters 1–3, start working from the summary of commands (chapter 13), looking up commands and concepts, I feel that chapter 4 is really also part of the introduction to TEX. Call it a higher introduction.

The following chapters 5–9 treat TEX commands, grouped by subject. Here too the explanations are clear, but they are less complete than I would like them. It has been a wise decision not to treat each command separately, but to tackle a few commands at a time, for instance `\hss` and `\vss`, or `\unskip`, `\unkern`, and `\unpenalty`.

Chapters 10–12 are probably a good selling point for this book: let it suffice that the titles are 'Tips and tech-

niques', 'Making sense of error messages', and 'A compendium of useful macros'. This last chapter contains an 'extended plain format', containing such much wanted macros as for cross references and left aligned display equations. Explanations of these macros limit themselves to explanations of the way to use them. A 'Capsule summary of commands' and an index complete the book.

On the whole, I find this book very clearly written, and all its information is readily accessible. However, I was a bit annoyed by the small errors that I found. For instance, the delimiters around `\..withdelims` commands don't grow as the authors claim; they are determined by font parameters 20 and 21 of the symbol font. Also, the remarks about the depth (height) of a `\vbox` (`\vtop`) on pages 52 and 161/2 are at odds; in principle this dimensions is the depth (height) of the last (first) box or rule. On page 52 it is stated that theses dimensions are zero if the last (first) item is kern or glue – which is wrong for the `\vtop`; on page 161/2 it is stated that it is zero if the last (first) item is not a box or rule – which is wrong for the `\vbox`. For the exceptions to these statements consider whatsits.

As I mentioned above, the question underlying this second part of the book started 'What kind of book would I need now, as a more experienced user'. By 'experienced user' the authors apparently do not mean an aspiring TEX hacker, since this book explains commands as such, but much less the large scale mechanisms connecting them.

For instance, one technique in chapter 10, 'Leaving space at the top of page', is treated in a mere five lines: the reader is told that `\vskip` does not work, but that `\topglue` does. I was particularly struck by this, since I didn't know the latter command, which is a late addition to TEX version 3. Neither here, nor in the systematic reference chapters is it mentioned whether this is a macro or a primitive. That information can only be found in the command summary; it is not even in the index, like it is in the TEX book.

Another example: page 86 states that 'When TEX breaks a page, it discards any sequence of glue, kerns, and penalty items that follows the break'. This is rather a simplification of what really happens; one might even say that this is simply not true. However, it is a convenient way of looking at things, and as long as you stick to the plain TEX output routine you never notice the difference.

The most obvious sign that the authors do not aim at TEX hackers is of course the fact that they repeatedly refer to the TEX book for the details. On page 167 it even says 'if you want to get adventurous you can learn all about it from pages [. . .] of the TEX book'.

In general, this book gives good factual information, and the information is very easy to find. What it lacks are the explanations, not of commands but of mechanisms. For instance, expansion is never treated as such. Also the problems of expansion and timing in macros for cross references and table of contents are not discussed.

But most people will not write such macros, and since some very handy macros are given in chapter 12, this book can be useful for people wanting to understand and modify or extend existing macros. And as an introduction, it is simply a good book.

BIJLAGE CC

An indentation scheme

Victor Eijkhout
University of Illinois at Urbana-Champaign
305 Talbot Lab
104 S. Wright Street
Urbana, Illinois 61801, USA
u641001@HNYKUN11

Indentation is one of the simpler things in TEX: if you leave one input line open you get a new paragraph, and it is indented unless you say `\noindent`. And if you get tired of writing `\noindent` all of the time, you declare

```
\parindent=0pt
```

at the start of your document. Easy.

More sophisticated approaches to indentation are possible, however. In this article I will sketch a quite general approach that can easily be incorporated in existing macro packages. For a better appreciation of what goes on, I will start with a tutorial section on what happens when TEX starts a paragraph.

1 Tutorial: paragraph start

When TEX is not busy typesetting mathematics, it is processing in *horizontal mode*, or *vertical mode*. In horizontal mode it is putting objects – usually characters – next to each other, in vertical mode it is putting objects – usually lines of text – on top of each other.

To see that there is a difference, run the following pieces of code through TEX:

```
\hbox{a}
\hbox{b}
\bye
```

and

```
a
\hbox{b}
\hbox{c}
\bye
```

You notice that the same objects are treated in two different ways. The reason for this is that TEX starts each job in vertical mode, that is, stacking material. In the second piece of input TEX saw the character 'a' before it saw the boxes. A character is for TEX the sign to switch to horizontal mode, that is, lining up material, and start building a paragraph.

Commands that can make TEX switch to horizontal mode are called 'horizontal commands'. As appeared from the above two examples characters are horizontal commands, but boxes are not. Let us now look at the two most obvious horizontal commands: `\indent` and `\noindent`.

1.1 `\indent` and `\noindent`

`\indent` is the command to start a paragraph with indentation. TEX realizes the indentation by inserting a box of width `\parindent`. If you say `\indent` somewhere in the middle of a paragraph you get some white space there, caused by the empty box.

`\noindent` is the command to start a paragraph without indentation. After this command TEX merely switches to horizontal mode; no indentation box is inserted. If you give this command somewhere in the middle of a paragraph it has no effect at all.

If TEX sees a horizontal command that is not `\indent` or `\noindent`, for instance a character, it acts as if the command was preceded by `\indent`. This is why paragraphs usually start with an indentation.

As an illustration here is a small variation on the above two examples:

```
\noindent
\hbox{a}
\hbox{b}

\indent
\hbox{a}
\hbox{b}
\bye
```

1.2 `\everypar`

TEX performs another action when it starts a paragraph: it inserts whatever is currently the contents of the token list `\everypar`. Usually you don't notice this, because the token list is empty in plain TEX (the TEX book [3] gives only a simple example, and the exhortation 'if you let your imagination run you will think of better applications'). LATEX [5], however, makes regular use of `\everypar`. Some mega-trickery with `\everypar` can be found in [2].

• Just to show how this works, I put in front of this paragraph the statement

```
\everypar={$\bullet\quad$}
```

That is, I told TEX that `$\bullet\quad$` should be inserted in front of a paragraph.

• There's nothing specified for this paragraph; I get the bullet for free, as `\everypar` does exactly what its name promises: it is inserted in front of *every* paragraph.

At the end of the previous paragraph I specified

```
\everypar={}
```

so nothing is inserted from this paragraph onwards.

1.3 Removing indentation

Every TEX user knows that indentation can be prevented globally by setting `\parindent` to zero. However, this is rather crude, and if you use the plain TEX macros you may notice several rather unpleasant side effects of this action, for instance when you use the macros `\item` and `\footnote`.

It is possible to use `\everypar` to prevent indentation, or more correctly: to remove indentation. This can be achieved by

```
\everypar={{\setbox0=\lastbox}}
```

This needs some explanation.

If the last item that was processed by TEX is a box, then that box is accessible by the command `\lastbox`. If the last item was not a box then `\lastbox` is an empty box, but no error ensues. As the `\everypar` list is inserted after any indentation box, the `\lastbox` command will get hold of the indentation box if there is one. By assigning the last box to another box register – here `\box0` – it is removed from where it was previously.

Finally, the statement

```
\setbox0=\lastbox\
```

is enclosed in braces. TEX's grouping mechanism restores values when the group ends that were current when the group began. In this case it has the effect of totally removing the indentation box: first it is taken and assigned to `\box0`, then the value of `\box0` is restored to whatever it was before the group began.

1.4 Other actions at the start of a paragraph

In the above discussion I have omitted one action that takes place at the start of a paragraph: TEX inserts (vertical) `\parskip` glue above the paragraph. As this has no relevance for the subject of indentation I will not go into it any further. However, in a subsequent article I will give more information about `\parskip`.

2 To indent or not to indent

In classical book typography [4] every paragraph is indented, with the exception of the first paragraph of a chapter. Nowadays a design where no paragraph indents is quite common. There are two mixtures between always indenting and never indenting: occasionally indenting, and occasionally not indenting. Thus it seems possible to characterize indentation strategies by two yes/no parameters: one that decides whether paragraphs should indent in principle, and another parameter that can overrule those decisions. Let us now see how this can be implemented in TEX.

2.1 Implementation

Above I have already indicated that changes to `\parindent` should be avoided. Let us then assume that `\parindent` is greater than zero, even if we will never indent a paragraph (see [1] for other uses for the `\parindent` quantity). We must then realize unindented paragraphs by removing their indentation as explained above.

First we need a macro for removing the indentation:

```
\def\removeindentation
    {{\setbox0=\lastbox}}
```

Then we need the switches that control indentation:

```
\newif\ifNeedIndent %as a rule
\newif\ifneedindent %special cases
```

Now for the definition of `\everypar`. This is a bit tricky.

Let us first collect some bits and pieces. The main question is to decide when `\removeindent` should be called. This is for instance the case if `\NeedIndentfalse`, and that parameter is not overruled by `\needindenttrue`.

```
\ifNeedIndent
    \ifneedindent
    \else \removeindentation
\fi \fi
```

Indentation should also be removed in case `\NeedIndenttrue`, but when that parameter is overruled by `\needindentfalse`.

```
\ifNeedIndent
\else \ifneedindent
      \else \removeindentation
\fi   \fi
```

Next we should make sure that `\ifneedindent` is used only for exceptional cases: if the user or a macro sets this parameter to a different value from `\ifNeedIndent`, then that should be obeyed exactly

once.

```
\ifNeedIndent
      \ifneedindent
      \else \needindenttrue \fi
\else \ifneedindent \needindentfalse
\fi   \fi
```

This is then the full definition of `\everypar`:

```
\everypar={\controlledindentation}
\def\controlledindentation
   {\ifNeedIndent
          \ifneedindent
          \else \removeindentation
                \needindenttrue
          \fi
    \else \ifneedindent
                \needindentfalse
          \else \removeindentation
    \fi   \fi}
```

Another implementation would be possible:

```
\def\controlledindentation
   {\ifneedindent
    \else \removeindentation \fi
    \let\ifneedindent=\ifNeedIndent}
```

This saves one conditional, but for most paragraphs it involves an unnecessary `\let` command.

2.2 Usage

My aim in developing this indentation scheme was to hide all commands pertaining to indentation in macros. The user should have to specify only once whether paragraphs should indent as a rule:

```
NeedIndenttrue
```

and then macros should declare the exceptions:

```
\def\section#1{...
    \needindentfalse
    ...}
```

2.3 But couldn't you simply . . . ?

Maybe people who read this have written macros themselves that end like

```
\def\section#1{...
    ...
    \noindent}
```

or

```
\def\section#1\par{...
    ...
    \noindent}
```

This works reasonably well, but it is not completely safe. In the first case there shouldn't be an empty line after a

```
\section{...}
```

call, and in the second case there can only be one empty line after

```
\section ...
```

The reason for this is that *every* empty line generates a `\par` command, which annuls the effect of the `\noindent`. Hence the more drastic approach.

An argument the other way around can also be found, by the way. As Ron Whitney pointed out to me, the following piece of code causes trouble:

```
\section{Title}%
  {\smallcaps The first} words are ...
```

Any changes made by `\everypar` are now effected *inside a group*. In this case one remedy is to insert a `\leavevmode` command, or to define

```
\def\smallcapswords#1{\leavevmode
   {\smallcaps #1}}
```

which can be used at any place.

Another remedy would be to let all assignments controlling indentation be global. However, there are some subtle objections to this.

3 About macro packages and users

Above I remarked that plain TEX does not use `\everypar`, and that LATEX redefines it a lot. This means that in plain TEX the user is free to take every value of `\everypar` that he or she likes; in LATEX every attempt of the user to use `\everypar` is immediately thwarted.

One might ask how the use of `\everypar` that I have sketched compares to this. Can the user be allowed to access `\everypar`, even if the macro package needs it all the time?

In my own 'Lollipop' format I have taken the following way out. The user or the style designer is allowed to fill in `\everypar`, as long the statement

```
\the\everyeverypar
```

is included. Here `\everyeverypar` is the token list with the constant actions such as indentation control that should be performed always.

A format designer who wishes to hide even this from the user or the style designer, could use the following piece of code

```
\newtoks\temppar
\def\everyparagraph
   {\afterassignment\xevpar
    \temppar}
```

```
\def\xevpar
   {\edef\act{\everypar=
        {\the\temppar
         \the\everyeverypar
        }}%
    \act}
```

so that it becomes possible to write

```
\everyparagraph={\DoSomething
    \everyparagraph={}}
```

while the `\everypar` will still contain all of the constant actions.

Short explanation: `\everyparagraph` is a macro that is made to look like a token parameter by the use of `\afterassignment`. This latter command sets aside the token `\xevpar` for insertion after the first assignment occurring; then whatever follows is assigned to `\temppar`. After this assignment the macro `\xevpar` unwraps the `\temppar` token list and the constant actions into `\everypar`.

4 Conclusion

In a systematic layout indentation commands need never be typed by the user; they can all be hidden in macros. Using `\everypar` it is possible to prevent indentation both in single instances, and throughout the document. This has the advantage that is is not necessary to zero the `\parindent` parameter or use `\indent` and `\noindent` instructions.

The approach of employing `\everypar` as sketched above can also be used for a paragraph skip scheme, as I will show in a subsequent article.

References

[1] Donald Knuth, The TEX book, Addison-Wesley Publishing Company, 1984.

[2] Leslie Lamport, LATEX, a document preparation system, Addison-Wesley Publishing Company, 1986.

[3] Victor Eijkhout, Unusual paragraph shapes, Tugboat vol. 11 (1990) #1, pp. 51–53.

[4] Stanley Morison, First principles of typography, Cambridge University Press, 1936.

[5] J. Braams, V. Eijkhout, N.A.F.M. Poppelier, The development of national LATEX styles, TUGboat vol. 10 (1989) #3, pp. 401–406.

BIJLAGE DD

A parskip scheme

Victor Eijkhout
University of Illinois at Urbana-Champaign
305 Talbot Lab
104 S. Wright Street
Urbana, Illinois 61801, USA
u641001@HNYKUN11

While I was working on the LaTeX styles described in [1], it became apparent to me that lots of people are rather fond of the sort of layout that can be described as

```
\parindent=0cm
\parskip=6pt % or other positive size
```

Unfortunately, most of them realize this layout by no more sophisticated means than simply inserting these two lines at the beginning of the input. The drawback of such a simple action is that all sorts of vertical spaces are augmented by the `\parskip` when there is absolutely no need to, or where it is positively unwanted. Examples of this are the white space below section headings, and the white space above and below list environments in LaTeX.

In this article I will present an approach that unifies the paragraph skip and the white spaces surrounding various environments. Since the macros given below make use of the `\everypar` token list, this article may be seen as a sequel to an earlier paper on an indentation scheme [2], which is based on a similar principle. The `\everypar` parameter was explained there.

\parskip

TeX starts a paragraph when it switches from vertical to horizontal mode. The vertical mode may have been initiated by a `\par` (for instance because of an empty line after a preceding paragraph) or by a vertical skip command; the switch to horizontal mode can be effected by, for example, a character or a horizontal skip command (see the list in [3, p. 283]). Immediately above the first line of the paragraph TeX will then add glue of size `\parskip` to the vertical list[1].

Apparently, then, the `\parskip` parameter is very simple to use. That this is only an apparent simplicity becomes clear in a number of instances.

For instance, unless precautions are taken, the white space below headings is augmented by the paragraph skip. Precautions against this are not particularly elegant; the easiest solution is to include a

```
\vskip-\parskip
```

statement, to backspace the paragraph skip in advance. Such an approach, however, is somewhat error-prone. Vertical spacing will be messed up if what follows is not a paragraph, but a display formula or a box.

Similar considerations apply to the amounts of white space that surround, for example, list environments, as in LaTeX.

Paragraph skip: to be or not to be

(This section is something of a footnote to the rest of the article. Readers who are not interested in layout consideration may skip the rest of it.)

Ordinarily in plain TeX and in LaTeX the paragraph skip is set to `0pt plus 1pt`, which gives pages some 'last resort' stretchability. However, even an amount of vertical space as small as one point may become very visible, and often without need (see for instance the first page of the preface of [3]).

Furthermore, Stanley Morison states that not indenting paragraphs is 'decidedly an abject method' [4]. However, reading his intention instead of his words, he is only concerned with the recognizability of the individual paragraphs. The positive value of the paragraph skip is sufficient to ensure this. If a layout is based on zero values for both `\parindent` and `\parskip`, one may for instance give the `\parfillskip` a positive natural width to prevent last lines of a paragraph from almost, or completely, lining up with the right margin.

Neither Donald Knuth nor Leslie Lamport seem to have given much thought to the case where the paragraph skip has a positive natural width. Leslie Lamport dismisses all potential difficulties with the remark that 'it is customary not to leave any extra space between paragraphs' [5, p. 94].

[1] Unless this paragraph is at the start of a vertical list, for instance at the start of a vertical box or insertion item.

Environments and white lines

Given that the paragraph skip appears to interact with explicit vertical spacing in user macros, it may seem like a good idea to find a unified approach to both. In the rest of this article I will describe the implementation of the following basic idea: *give the paragraph skip the value zero whenever you do an explicit vertical skip*.

For the presentation I assume a context with some form of environments. These are the assumptions that I make about such environments:

- An environment is a portion of material that is vertically separated from whatever is before and after it. Thus, according to this definition, a portion of a paragraph cannot be an environment, nor can an environment start or end in the middle of a paragraph.
- An environment has associated with it three glue parameters: to an environment `foo` correspond `\fooStartskip` (glue above the environment), `\fooParskip` (the paragraph skip inside the environment), and the `\fooEndskip` (glue below the environment).
- At the outset of the environment a
  `\StartEnvironment{foo}`
  statement is executed; at the end of the environment a macro
  `\EndEnvironment{foo}`
  is executed. These statements are assumed to contain a `\begingroup` and `\endgroup` respectively.

Such assumptions are sufficiently general for the macros below to be adaptable to existing macro packages. At first sight it would appear as if section headings are not covered by the above points. However, there is no argument against the start and end of an environment occurring in the same macro.

Tools

First I will present two auxiliary macros: `\csarg` and `\vspace`.

The command `\csarg` is only needed inside other macros; it is meant to enable constructs such as

```
\csarg\vskip{#1Parskip}
```

Its definition is

```
\def\csarg#1#2{\expandafter
    #1\csname#2\endcsname}
```

By way of explanation of this macro, consider a simple example. Let us assume that there exists a macro

```
\def\startlist#1{ ...
    \csarg\vskip{#1Startskip}
    ...}
```

The call

```
\startlist{enumerate}
```

will then lead to the following call to `\csarg`:

```
\csarg\vskip{enumerateStartskip}
```

This expands to

```
\expandafter\vskip
 \csname enumerateStartskip\endcsname
```

Now the `\expandafter` forces `\csname` to be executed before the `\vskip`, so the next step of the expansion looks like

`\vskip\enumerateStartskip` and this statement can simply be executed.

Next I need a generalization of `\vskip`, which I will call `\vspace`: a number of calls to `\vspace` should have the effect that only the maximum argument is placed.

```
\newskip\tempskipa
\def\vspace
 #1{\tempskipa=#1\relax
    \ifvmode \ifdim\tempskipa<\lastskip
        \else \vskip-\lastskip
              \vskip\tempskipa
        \fi
    \else \vskip\tempskipa \fi}
```

This may need some explanation too. First, by the assignment

```
\tempskipa=#1
```

I allow the argument of `\vspace` to be both a control sequence, for instance `\parskip`, and a denotation, for instance `5pt plus 3pt`. If one omits the assignment, the latter option would cause trouble in the `\ifdim` test.

The decision to keep the maximum value of the skip, instead of always replacing the last skip, was motivated by phenomena such as a display formula at the end of a list. If the skip below the display is larger than the vertical glue below the list (which may for instance be zero), the former should be retained entirely.

Note that this macro can insert vertical space of size zero. This will for instance happen if it follows a `\par` command at the end of a paragraph. It is then called in vertical mode, but the last item on the vertical list is a box (the last line of the paragraph) instead of a glue item. The parameter `\lastskip` will then be zero.

The environment macros

In this section, I will give the implementation of the macros `\StartEnvironment` and `\EndEnvironment`.

There is a remarkable similarity between these two macros. As I explained above, the basic idea is to have only explicit spacing above and below the environment; thus, the value of `\parskip` should then be zero both for the first paragraph in the environment, and for the first paragraph that follows it. Both macros should then

- save the current value of the paragraph skip;

- set the paragraph skip to zero;
- give TEX a signal that, somewhere in the near future, the old value of `\parskip` is to be restored.

For this I allocate a skip register and a conditional:

```
\newskip\TempParskip
\newif\ifParskipNeedsRestoring
```

The basic sequence for both macros is then

```
\TempParskip=\parskip
\parskip=0cm\relax
\ParskipNeedsRestoringtrue
```

For both macros, however, this sequence needs to be refined slightly.

The paragraph skip to be 'restored' at the start of the environment is the specific value associated with that environment. This gives us:

```
\def\StartEnvironment
 #1{\csarg\vspace{#1Startskip}
    \begingroup %% make changes local
    \csarg\TempParskip{#1Parskip}
    \parskip=0cm\relax
    \ParskipNeedsRestoringtrue}
```

Note that the statement

```
\csarg\TempParskip{#1parskip}
```

denotes an assignment (without an optional equals sign) here.

At the end of the environment we have to be more careful. It may be the case that the environment being ended was inside another environment, and occurred before the first paragraph inside that environment. In that case the value of `\parskip` is zero, and the proper value must still be restored. Therefore, no further actions are required. We arrive at the following implementation:

```
\def\EndEnvironment
 #1{\csarg\vspace{#1Endskip}
    \endgroup %% restore global values
    \ifParskipNeedsRestoring
    \else \TempParskip=\parskip
          \parskip=0cm\relax
          \ParskipNeedsRestoringtrue
    \fi}
```

Note that both macros start with a vertical skip. This prevents the `\begingroup` and `\endgroup` statements from occurring in a paragraph. On a side note: since these macros are executed in vertical mode, I have not bothered to terminate any lines with comment signs. Any spaces generated by these macros are ignored in vertical mode.

Paragraph skip restoring

So far, I have ignored one important question: how exactly is restoring the paragraph skip implemented. For this I use the `\everypar` token list. Basically then, the idea is the same as in [2]: the occurrence of a paragraph will automatically have TEX perform, through the insertion of the `\everypar` tokens, the actions necessary for subsequent paragraphs.

```
\everypar=
   {\ControlledIndentation
      %see Tugboat 11, #3
    \ControlledParskip}
\def\ControlledParskip
   {\ifParskipNeedsRestoring
       \parskip=\TempParskip
       \ParskipNeedsRestoringfalse
    \fi}
```

The cost of a controlled paragraph skip is then one conditional per paragraph. Conceivably, this cost could even be reduced further (to almost zero) by defining

```
\def\CPS % Controlled Parskip
   {\ifParskipNeedsRestoring
       \parskip=\TempParskip
       \ParskipNeedsRestoringfalse
       \let\ControlledParskip=\relax
    \fi}
```

and including a statement

```
\let\ControlledParskip=\CPS
```

in both the `\StartEnvironment` and `\EndEnvironment` macros, and at the start of the job (for instance by including it in the macro package). This approach, however, does not particularly appeal to me. Too much 'pushing the bit around'.

Conclusion

The `\parskip` parameter is arguably the most tricky parameter of TEX. Its workings are very easy to describe, but in actual practice difficulties arise. In this article I have described how treatment of the paragraph skip can be integrated with the glue above and below environments. As in an earlier article on indentation, I use for this the `\everypar` parameter as an essential tool.

References

[1] J. Braams, V. Eijkhout, N.A.F.M. Poppelier, The development of national LATEX styles, TUGboat vol. 10 (1989) #3, pp. 401–406.

[2] Victor Eijkhout, An indentation scheme, TUGboat vol. 11 (1990) #4.

[3] Donald Knuth, The TEX book, Addison-Wesley Publishing Company, 1984.

[4] Stanley Morison, First principles of typography, Cambridge University Press, 1936.

[5] Leslie Lamport, LATEX, a document preparation system, Addison-Wesley Publishing Company, 1986.

BIJLAGE EE

SGML (, TEX and . . .)

C.G. van der Laan

September 1990

Abstract

What SGML (and TEX) is all about is given in a nutshell. Markup of example document elements, by SGML and LATEX, are provided. Coupling SGML to TEX is considered by direct translation and by the intermediate procedural markup phase. Interfacing SGML to (La)TEX is also addressed. Some guidelines are provided in order to decide when SGML, or TEX (alone, both, or neither) might be beneficial. It is a 3-in-1 paper: what is SGML and TEX all about, examples of marked up copy in SGML and (La)TEX and the coupling issues, finished up with a literature compilation.

What is a Document?

The Association of American Publishers (AAP) *Reference Manual on Electronic Manuscript Preparation and Markup* defines a document as:

> A document is an organized collection of smaller pieces of text (such as chapters) and images (such as figures) that are called elements. The elements in a document have a relationship to each other which gives the document a definite organization called document structure.

Lifecycle-phases of documents

Manuscript preparation requires that additional information be interspersed within the text to aid any subsequent processing. That information, called markup, is usually specific to a particular publisher, system or printing device. A universal standard method of of marking up electronic manuscripts, however, offers many advantages.

The *complete Lifecycle* of a document can be thought of as:

- preparation,
- distribution,
- reading,
- storing (Paper? Electronically? Optically?),
- other usage, reuse?[1]

Standard Generalized Markup Language (SGML) supports the complete Lifecycle, where *future* usage of the document is not necessarily restricted to printing. SGML must be complemented, however, by generally accepted Document Type Definitions (DTDs). The Association of American Publishers [AAP,1987 and 1989] and the British National Bibliography [Smith, 1987] have provided some DTDs. In order to serve the primary aim of publishing coupling to formatters must be supported too.

TEX supports formatting and electronic document exchange.

What is SGML?

SGML provides us a language to describe documents. SGML has it made possible to achieve two goals:

1. establish a standard means of identifying and tagging parts of an electronic manuscript so that computers can differentiate between these parts; and
2. provide some logical ways of representing special characters, symbols and tabular material, using only the ASCII character set usually found on standard keyboards.

SGML is defined in ISO8879 [1986]. An introduction for SGML is given by Barron [1989], a gentle introduction is included in Sperberg-McQueen & Burnard [1990]. Introductory books are Bryan [1988] and van Herwijnen [1990].

Purpose

The purpose of SGML is to facilitate *information* exchange, and reusability (in other contexts, even yet unknown contexts),

—Then and There—

via a description *language*, where information is packed in documents, containing, text, graphics, formulas, tables, etc.

[1] We can talk about reuse and rework. From the author's point of view we are dealing with rework. Publishers like to reuse copy.

Meta Language

SGML is a *meta* language, which can be used to define an arbitrary number of markup languages in a standardized way. This means, for any class of documents, markup rules can be prescribed by SGML, yielding a *language*, the Document Type Definition, for that class. The parser checks compliance of the marked up copy to the DTD.

Standard

Formerly: no consensus on markup "codes" (WordPerfect, Wordstar, MacWrite, . . . ; Scribe, troff, TEX, LATEX, . . .)

Presently: SGML ISO standard

Standard $\stackrel{\text{def}}{=}$ It can be used to define an arbitrary number of markup languages in a *standardized* way.

Entails:
general applicability,
longer longevity,
improved reusability,
enhanced exchange possibilities.

Generalized

Formerly: (typeset) *marks* for *specific* "here and now" printers, via *direct* markup.

Presently: Marks are *generic*.[2] This is done with procedural markup. Macro calls are inserted as markup tags, where the implementation of the macros (the format or style file) represents the style, accounts for the fonts, etc. The printer hardware is shielded by intermediate languages. Intermediate language copy is printed via drivers. Change of style needs another style file, no modification of tagged copy is necessary. Change of printer hardware needs another driver, no modification of tagged copy nor modification of format file!

Generalized $\stackrel{\text{def}}{=}$ Abstraction from the specific (printing) to the general (other usage), by emphasizing the *structure* of a document and to specify intent without regard for appearance.

Markup

Formerly: (typeset) *marks* in the margin (Marks are bound to a version; no "data-integrity")

Presently: Marks are integrated along with copy (Data-integrity of markup code is preserved.)

Markup $\stackrel{\text{def}}{=}$ Term used to describe codes added to the electronically prepared document.

Author's point of view

Authors have to markup their copy with

1. awareness of the DTD which applies to the document; either the DTD must be understood or templates must be available;
2. knowledge of which (begin) tags to use where and how;
3. knowledge of ranking, attribute use, tag minimization;
4. knowledge about how to create entity references.

These aspects are treated in author's guidelines. The above can be alleviated by providing an SGML *environment*, or better, a document preparation environment, supported by menus and templates with prompts. I agree with the general expectation that authors can concentrate on structure and content by using a standard (generic) markup language, or a sufficiently advanced document workbench, leaving formatting issues to publishers, or software vendors. Because of this "separation of concerns" the author's task is simplified.

Publisher's point of view

Publishers make use of sufficiently accepted DTD s. They provide authors with guidelines and proof tools. DTD writing requires knowledge of the various types of markup, such as presentational, direct, procedural and descriptive markup.

Example markups

No markup

In order to remind the unpleasant look of documents with just words, we start with a no markup example

```
TeX A system for formatting text
TeX and the accompanying macro
package LaTeX provide powerful means ...
```

Presentational markup

Documents with tabs, indentations, and in general positional control make use of what is called presentational mark up, in order to convey the meaning

```
TeX:
A system for formatting text.

  TeX and its accompanying macro
package LaTeX provide
powerful means of formatting
text to be output on either
  - a simple matrix printer,
  - a laser printer or
  - a photo typesetter.
```

Presentational markup is functional with poetry, such as Alice's mousetail as mentioned by Malcolm Clark [1989] or DEK's favourite poem of Piet Hein [*The Errors of TEX*, 1989], where the words are arranged along an ellipse.

[2] Not specific to print/plot/phototypesetter hardware.

Direct markup

When specific print instructions are included, we get direct markup:

```
@T:                          []
A system for formatting text.[]
                             [I]
@T and its accompanying macro
package @LT provide
powerful means of formatting
text to be output
on either                    [I]
 - a simple matrix printer,  [I]
 - a laser printer or        [I]
 - a photo typesetter.
```

`[I]` is a print instruction indicating to go to the next line and *indent*; @<name> stands for a process with special format effect.

Procedural (LATEX) markup

A markup command, where the implementation of the command contains print instructions, is considered a procedural markup command; when the printer is changed the implementation has to be changed too, not the marked up copy. LATEX markup of the example reads

```
\subsection*{\TeX}
A system for formatting text.
\par
  \TeX\ and its accompanying macro
  package \LaTeX\ provide
powerful means of formatting text
to be output on either
\begin{itemize}
\item simple matrix printer,
\item a laser printer or
\item a photo typesetter.
\end{itemize}
```

Descriptive (SGML) markup

Descriptive markup goes even further and uses markup which describes the structure and intent of the various parts of the document:

```
<h>&TeX;
<p>A system for formatting text.
<p>&TeX; and its accompanying macro
   package &LaTeX; provide
   powerful means of formatting
   text to be output on either
<li>
<it>simple matrix printer,
<it>a laser printer or
<it>a phototypesetter.
</li>
```

Formatting information with SGML?

It is possible to convey formatting information via SGML. This is done with element attributes or with Processing Instructions. Other symbols than those in the ASCII character set are often denoted by an entity reference to the font containing those symbols, with appropriate loading of the font elsewhere. With respect to attributes, one can think of specifying open space in order to include illustrations from other (electronic) sources. Also, indication of a representation choice is possible if properly accounted for in the DTD. Consider for example the representation of labels of list items: alphabetical or roman/arabic numeral.

```
<li number=alpha>
<it>a simple printer,
<it>a laser printer or
<it>a photo typesetter.
</li>
```

It is possible in SGML to include document parts which already contain format information. The parser must be told to lay back. For a notation to be allowed it must be declared via e.g.

```
<!NOTATION TeX SYSTEM>
```

for TEX formatted copy. Appropriate entity and attribute specifications are also needed in the DTD. For authors the equation formatting with the type attribute (value=TEX) has to be supplied as:

```
<eqn type=tex>
  $$X\cap(A\cup B)=(X\cup A)\cap(X\cup B)$$
</eqn>
```

Processing Instructions (PI s) can be used to tell the local system how it should process data contained within a document. For example, SETM typography markup instructions:

```
<p><?[s24][sec][rm]>T<?[pri][rm]his>
           paragraph ...
```

In this case the SETM instructions are in brackets, preceded by the PI open delimeter <?. The meaning of this instruction is to treat "**T**" as "**24**pt" initial letter to be set using the **rom**an version of the face currently defined in the **sec**ondary type family.

Availability

As mentioned by Herwijnen [1990], Sobemap and The Publisher are some SGML systems that are already available.

Support

Support for SGML is done by the companies, as part of automation projects. There also exists a Dutch chapter of the SGML Users Group.[3]

[3] SGML-Holland secretary: D. van Wijnen, Wolters Kluwer. P.O. Box 989, 3300AZ Dordrecht. 078–334933; e-mail: surf003@kub.nl.
SGML User's Group secretary: S. G. Downie, Softquad Inc, 720 Spadina Avenue, Toronto, Ontario M5S 2T9, Canada.

Courses

Courses are also provided by private companies, and the National Normalization Institutes.

What is SGML not?

SGML is not

- WYSIWYG (pronounced wĭsēwĭg, and stands for what you see is what you get),
- a formatter, certainly not a standard formatter.

What is TEX?

TEX stands for $\tau\varepsilon\chi$, the first three letters of the Greek word for *technique*, which also means art. TEX is a *machine independent* formatting language designed by Don Knuth, [*The TEXbook*, 1990 (Version 3.0)]. Michael Doob gives an easy start to TEX in *A Gentle Introduction to TEX* [1989]. There also is an introduction in French by Seroul [1989] and various German introductions by Appelt [1988], Schwarz [1989]. Von Bechtolsheim [1990] is impressive. LATEX, by Leslie Lamport [1985], is a macro collection for simplified use of TEX. LATEX uses *procedural* markup. Buerger [1990] gives a LATEX introduction. Bruin [1989] gives a Dutch introduction to LATEX.

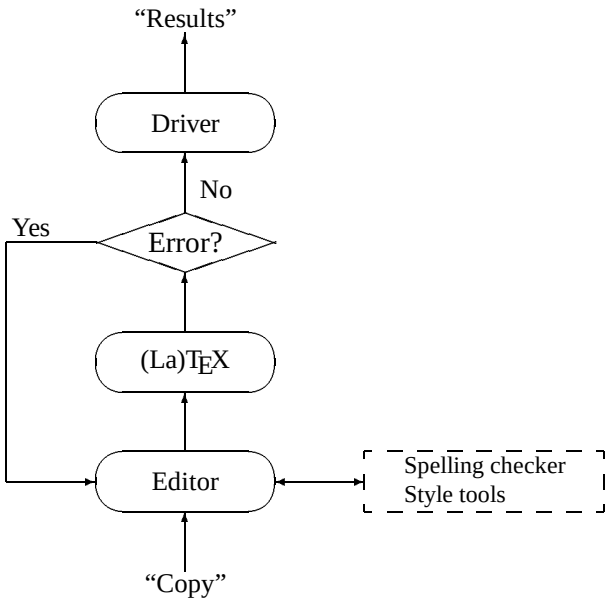


Figure 1: Correction cyclus

Purpose

The purpose of TEX is "making beautiful books."

Processing (La)TEX

LATEX is processed in three steps: edit the copy, format the copy to create a `dvi` file, and print the resultant `dvi` file. A diagram of this looks like:

copy $\overset{editor}{\rightarrow}$ `ASCII` $\overset{(La)TeX}{\rightarrow}$ `dvi`-file $\overset{driver}{\rightarrow}$ results

The more steps to the process, the more cumbersome is correction handling because of larger "loops." This is the case when the use of TEX is combined with SGML markup. The SGML parsing and linking extends the loop.

Availability

TEX is available on many computers under various operating systems with a variety of drivers for previewing (such as VDU), printing, and phototypesetting. Documents written in TEX or LATEX can be ported easily. Exchanging documents via e-mail is also generally possible except for the incorporated graphics. When graphics are part of the document, TEX can be combined with Postscript, which is used within the TEX community. TEX is in the *public domain.* Drivers are not. They do, however, generally have added value from the companies you buy the driver from. See the ads in any *TUGboat.*

Moreover, TEX systems can make use of fonts from various sources, such as Adobe's PostScript fonts and, of course, METAFONT.

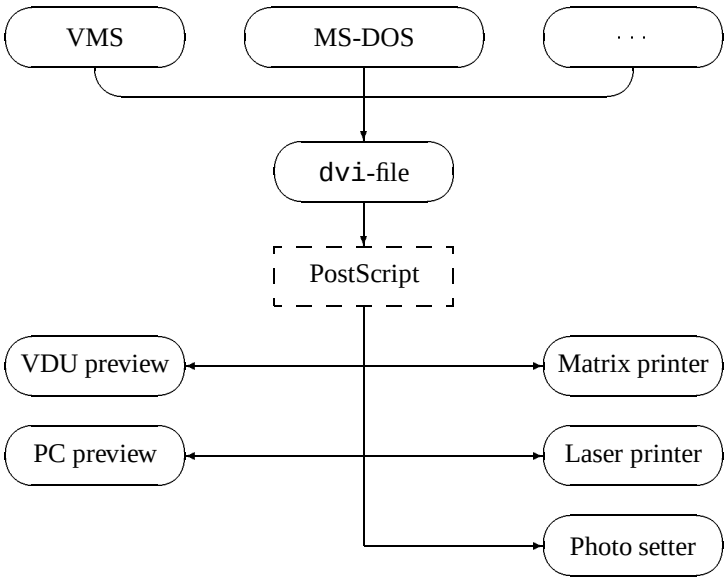


Figure 2: (La)TEX's use

Support

Support is organized by the various users groups. Software, style files, macros etc. are distributed (via e-mail, ftp or floppy disks) by the TEX Users Group (TUG)[4]; in the Netherlands it is distributed by NTG[5]; in France by GUTenberg; in Germany by DANTE; in the United Kingdom by ukTEXug; in the Nordic countries by "Nordic TuG". There is also the new TUGlib service in Utah.

Courses

Courses are organized by TUG and other TEX user groups, especially in conjunction with their main meetings.

Relationship: DTDs, SGML, TEX, formats and . . .

The relationship of TEX, SGML and other applications is illustrated in figure 3. An integrated[6] implementation is Arbortext's "The Publisher," which has AAP's DTD s built-in, and requires SUN hardware.

Note that the two backarrows denote some of the work in progress by Elsevier Science Publisher, Bleeker [1989], Poppelier [1990].

SGML ór TEX sufficient?

NO, needed are format files and DTDs as well! If a manuscript is printed with TEX for personal use only, there is nothing to worry about. When no reuse of a document is in sight, but remote publishing and electronic exchange are the case, it pays to use standard format/style files—which reflect the lay-out of the document type—along with TEX. When reuse or abstract structuring are being used, standard DTDs will be crucial. [7]

Interfacing vs. transformation

Interfacing copy marked up by any formatter to SGML can be prescribed in SGML via the NOTATION mechanism. Of course it has to be incorporated into the DTD and appropriately implemented: the parses should lay back and leave the formatting to the TEX marked up copy to TEX.

Transformation SGML into TEX is different. Using TEX as a back-end formatter to SGML can be done. It is supported by the link mechanism of SGML. Needed is at least a table of corresponding notations in order to substitute the markup tags from SGML into TEX. The other way round has to be done by a separate program. In the sequel we will study example document elements marked up by SGML and (La)TEX; transformation issues will be addressed as well.

[4] Editorial and TUG address: TEX Users Group, P.O. Box 9506, Providence RI 02940, USA. email: TUGboat@Math.AMS.com.

[5] NTG: Nederlandstalige TEX Gebruikersgroep. Secretary: G.J.H. van Nes, Postbus 394, 1740AJ, Schagen, 02246–4185; e-mail: vannes@ecn.nl, ntg@hearn.

[6] Ikons user interface, SGML layer, TEX layer, Postscript handling (optionally); with SGML, TEX and dvi files as intermediate results

[7] The meta-ness is a strong point —flexibilty, adaptabilty and openendedness— but, surprisingly at the same time it acts as a weak point —everybody writes or modifies DTD , with discrepancy as result.

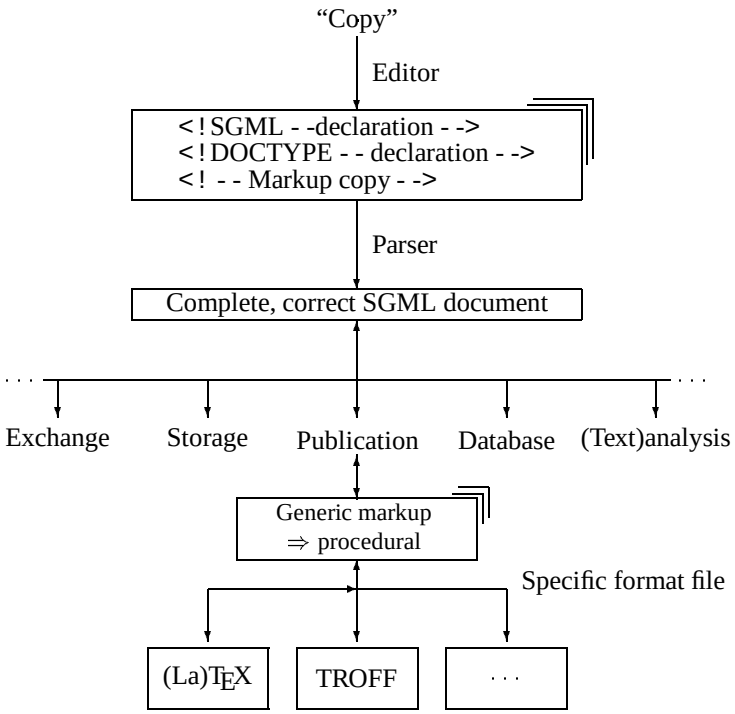


Figure 3: Relation SGML and (La)TEX

Examples

Simple text

As an example let us take the simple text given earlier. The (basic) SGML markup and LATEX markup have been given in previous sections. Coupling comes down to a change of representation, except for the omitted endtags. This direct approach needs the substitutions:

SGML	⇒ TEX
`<h>`	`\section{`
`<p>(first)`	`}`
`<p>`	`\par or blank line`
`<li>`	`\begin{itemize}`
`</li>`	`\end{itemize}`
`<it>`	`\item`
`&TeX;`	`\TeX`
`&LaTeX;`	`\LaTeX`

This suggests systematic coupling of all entities and tags to equivalents in LATEX. The explicit endtags are more natural to handle than the omitted ones. The handling of the first and latter occurences of `<p>` have to account also for respectively finishing the heading `</h>` and ending a paragraph `</p>`, which have been omitted.

Letter

A typical letter has the structure:

- Background
  Heading (Logo, address, phone, . . .)
  Footer (numbering, . . .)
- Context (running heads next pages, . . .)
- Reference
- Your reference
- Date
- Addressee (name, company, address, zip code)
- Beginning (Dear. . .)
- Contents
- End matter (Salutation, name, position)
- Additions (PS, enclosure, cc)

SGML markup

The SGML markup for a typical letter might look something like:

```
<!DOCTYPE letter PUBLIC
 -- DTD to be used  --
  "-//NTG//DTD Letter//EN">
<letter -- start-tag -->
<ref> CGL/Ba/B89-007
<yourref> MC/Ll/L89-001
<date> 4 august 1989
<address> Malcolm Clark, ICRF
<email>
     malcolm@icrf.ac.uk
</email>
<dear>Malcolm
<p> Thank you very much ...
...
<p> Some details about the course ...
...
<signed name=CGL>
</letter -- end-tag -->
```

LATEX specification

The same letter using LATEX markup might look like:

```
\documentstyle[12pt]{letter}
  \address{% return address
          C. G. van der Laan   \\
          \ldots}
  \signature{Kees}
\begin{document}
\begin{letter}{% address
          Malcolm Clark       \\
          \dots}
% no ref or your ref
% date is handled automatically
\opening{Dear Malcolm}
\par
Thank you very much \ldots
\begin{quote}
$\vdots$
\end{quote}
Some details about the course \ldots
\begin{quote}
$\vdots$
\end{quote}
\closing{Best regards}% Handles signature
%ps, cc, enclosure all possible
\end{letter}
\end{document}
```

Letter result

Because a sample LATEX letter could not be processed simultaneously in this paper, the printed letter has been omitted.

Transformation

What comes to mind when looking at both representations of marked up copy is the difference in the sequence order of tagged items in SGML and LATEX. With *complete* marked up SGML copy to be transformed into procedural TEX, there is no problem: in TEX macros strings can be stored for later usage. This will be shown along with the tabular example in the section transformation revisited.

Bridge card deal

In *TUGboat* 11#2 I have described typesetting bridge using TEX.

SGML markup

The SGML markup for Figure 4 might look like:

```
<deal><vuln>N/None
      <comm>Deal: demo
<hand n><spades>J74
        <hearts>AJ
        <diams> QJT2
        <clubs> Q874
<hand e><spades>K86
        <hearts>T9542
        <diams> 874
        <clubs> T3
<hand s><spades>QT952
        <hearts>Q83
        <diams> AK5
        <clubs> A6
<hand w><spades>A3
        <hearts>K76
        <diams> 963
        <clubs> KJ952
</deal>
```

(La)TEX specification

The procedural TEX markup for Figure 4 might look like:

```
\crdima{N/None}{\vtop{\hbox{Deal:}
                      \hbox{demo}}}%
  {\hand{J74}{AJ}{QJT2}{Q874}}%N
  {\hand{K86}{T9542}{874}{T3}}%E
  {\hand{QT952}{Q83}{AK5}{A6}}%S
  {\hand{A3}{K76}{963}{KJ952}}%W
```

Transformation

The transformation comes down to

SGML	⇒ TEX
`<deal>`	`\crdima`
`<vuln>N/None`	`{N/None}`
`<comm>DEAL: demo`	a suitable `\vtop`
`<hand x>`	`{\hand`

and all the cards per colour surrounded by curly braces, with an extra "`}`" after the clubs. Although once again a simple example, the translation table is not natural.

TEX macros

Figure 4 is created by using the TEX macros:

```
\def\hand#1#2#3#4{%
%Example: \hand{AKJ765}{AK9}{--}{T983}
\vtop{\hbox{\strut\s\enspace#1}
\hbox{\strut\h\enspace#2}
\hbox{\strut\d\enspace#3}
\hbox{\strut\c\enspace#4}}%end \vtop
}%end \hand
%
\def\crdima#1#2#3#4#5#6{%
%purpose: layout bridge hand
%#1 left upper text
%#2 right upper text
%#3, #4, #5, #6: N, E, S, W hands
\vbox{\halign{              &##\quad\cr
            #1&          #3&     #2\cr
 $\vcenter{#6}$&$\vcenter{\copy\NESW}$&
                       $\vcenter{#4}$\cr
               &          #5&       \cr
             }%end \halign
      }%end \vbox
}%end \crdima
%
\def\NESWfig{%
\vbox{\font\small=cmr9
\def\str{\vrule height2.2ex%
```

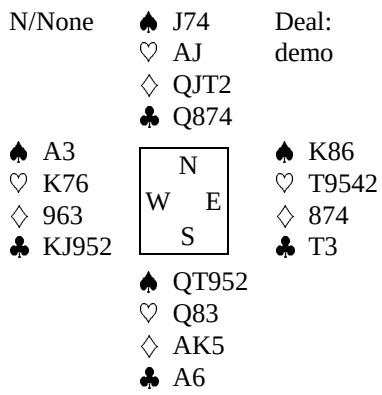


Figure 4: Bridge deal

```
    depth.75ex width 0pt}
\offinterlineskip\tabskip0pt\hrule
\halign{\vrule\hskip2pt\relax
##\hfil\tabskip3pt&  \str\hfil##\hfil&
##\hskip2pt\relax\hfil\vrule
                        \tabskip0pt\cr
  &      \hbox to 0pt{\hss\N\hss}&  \cr
\W&                    \phantom{N}&\E\cr
  &  \str\hbox to 0pt{\hss\S\hss}&  \cr
       }%end \halign
\hrule}%end \vbox
}% end \NESWfig
\setbox\NESW\hbox{\NESWfig}
```

SGML requirements

The following DTD is needed:

```
<!ENTITY % ISOpub PUBLIC
 "ISO 8879-1986//ENTITIES Publishing//EN">
<!ELEMENT deal - - (vuln, comm?, hand+)>
<!ELEMENT (vuln|comm) - O CDATA>
<!ELEMENT hand - O (spades,
                    hearts, diams, clubs)>
<!ATTLIST hand nesw (n|e|s|w) #REQUIRED>
<!ELEMENT (spades|hearts|diams|clubs)
                    - O CDATA>
```

Note. In the DTD we could have imposed sequence ordering by changing hand+ into (handn, hande, hands, handw). But this requires that all the hands are needed and that is too restrictive.

Some math

The following examples of mathematical formulas are borrowed from the "Mathematical Formulas" report [van der Laan, Coleman, Luyten, 1989]. In this report, SGML and LaTeX markup are supplied for formulas from various fields: elementary mathematics, set theory, geometry, functional analysis, calculus (differential equations, special functions, continued fraction), statistics, algebra (tensor calculus), homology (diagrams) and quantum mechanics. A few simple ones are selected here.

LaTeX results

The following was formatted with LaTeX markup:

$$X \cap (A \cup B) = (X \cup A) \cap (X \cup B)$$

$$x \notin A \not\subset B$$

$$\|a(x+y)\| \leq |a|.(\|x\| + \|y$$

$$\int \frac{1}{\sqrt{1+x^2}}\,dx = \log(1+\sqrt{1+x^2})$$

(Basic) SGML markup

To accomplish this with SGML markup, you might enter:

```
<fd>X&cap;(A&cup;B) =
     (X&cup;A)&cap;(X&cup;B)</fd>

<fd>x&nisin;A&nsub;B</fd>

<fd><fen d>a(x+y)<rp d</fen>&le;
  |a|.(<fen d>x<rp d</fen>
      +<fen d>y<rp d</fen>)
</fd>

<fd><in><opd><fr>1</><rad>1+
  x<sup/2/</rad></fr>dx</in>=
  <rf/log/(1+<rad>1+x<sup/2/</rad>)
</fd>
```

The DTD used is an adapted version of AAP's DTD by D.C. Coleman.

(Direct) TeX markup

LaTeX and TeX markup are very similar for these examples:

```
X\cap(A\cup B) =(X\cup A)\cap(X\cup B)

x \notin A \not\subset B

\|a(x+y)\| \leq |a|.(\|x\|+\|y\|)

\int\bfr1</>\sqrt{1+x^2}\efr dx =
                \log(1+\sqrt{1+x^2})
```

Transformation

To accomplish the SGML to TEX transformation, some general substitutions are needed:

SGML	⇒ TEX
<fd>	\[or $$
</fd>	\] or $$
<sup/2/	^2
etc.	

For the first set theory example the following substitutions are additionally needed

SGML	⇒ TEX
∩	\cap
∪	\cup

Functional analysis required moreover

SGML	⇒ TEX
<fen d>	|
<rp d>	|
≤	\leq

For the integral the following definition (format) is needed for the fraction, where use is made of </> as parameter separator:

```
\def\bfr#1</>#2\efr{{#1\over#2}}
```

Also needed are the substitutions

SGML	⇒ TEX
<in>	\int
<opd>	\relax
<fr>	\bfr
</fr>	\efr
<rad>	\sqrt{
</rad>	}
<rf/log/	\log

Note. A translation table is once again not straightforward; unnatural are the SGML difference in norm open and closing, and the fancy use of </>, i.e. null endtag for numerator and omitted opening tag for denominator. The short refence for the modulus sign is neat.

(Complete) SGML markup

The sobemap parser yielded the following (visually edited) result for the set theory example:

```
<FD DCN="GEO.FORM">
<FL>
X&cap;<FEN STYLE="S" LP="PAR">
        A&cup;B
      <RP STYLE="S" POST="PAR"></FEN>
=
<FEN STYLE="S" LP="PAR">
   X&cup;A
<RP STYLE="S" POST="PAR"></FEN>
   &cap;
<FEN STYLE="S" LP="PAR">
   X&cup;B
<RP STYLE="S" POST="PAR"></FEN>
</FL>
</FD>
```

This shows that complete SGML is verbose. For example, consider the complete tags for parenthesis "(" and ")". Thanks to the short reference mechanism the input can look natural. Coupling of the above to TEX can be done. How to handle automaticly attributes in the best way is not yet clear to me. It is not efficient for parenthesis, "(" and ")", to be expanded first by the parser into a fence tag with the appropriate attribute value, followed by substitution at the TEX level into "(" and ")" again.

Because matrices are treated similarly to tabular material, we have omitted a matrix example, and refer to the next sections, where a table is studied.

AAP's simple table

A simple table is characterized by: simple table entries, one header row, no header subrows, no footer, no intra referencing, and no caption. From the SGML technical point of view no attributes are used. AAP's example simple table [Markup of Tabular Material, 1989], even more simplified, is reproduced in figure 5.

(Basic) SGML markup

The SGML markup for Figure 5, with a minor structural adaptation and some layout modifications, reads:

```
<tbl>
<no>Table  AAP
<tt>Job Changes: 1973&ndash;1980
<th>
<th>Gain/Loss of Hospitals since 1973
<th>Total No. of CEO Job Changes
    1973&ndash;80
<th>Survival Rate of CEO's
<bdy>
@Texas|20||22%
@Maryland|5|42|24%
<ft>
<au>David Kinzer
<atl>Turnover Of Hospital Chief
     Executive Officers: A Hospital
     Association Perspective
<nme>Hospital and health Services
     Administration
<dt>May&ndash;June 1982
</tbl>
```

A DTD for simple tables is not separately provided by AAP; it is incorporated as part of the complex table DTD. The "simple table"-example obeys the following SGML structure description

```
<!ENTITY row    STARTTAG "row"          >
<!ENTITY column STARTTAG "c"            >
<!ELEMENT tbl     - - (hd, bdy, ft)     >
<!ELEMENT hd      - O (no?, tt?, th+)   >
```

Table AAP Job Changes: 1973–1980

	Gain/Loss of Hospitals since 1973	Total No. of CEO Job Changes 1973–80	Survival Rate of CEO's
Texas	+20	—	22%
Maryland	+ 5	42	24%

Source: David Kinzer, "Turnover Of Hospital Chief Executive Officers: A Hospital Association Perspective," *Hospital and health Services Administration* May–June 1982.

Figure 5: AAP's simplified table

```
<!ELEMENT bdy     - O row+              >
<!ELEMENT row     - O c+                >
<!ELEMENT ft
      - O (au|src|atl|nme|dt)           >
<!ELEMENT
      (th, c, au, src, atl, nme, dt)--
      - O (%t.cs;) -- Character string-->
<!SHORTREF tablemap "@" row
                    "|" column          >
```

Note. Some tags are presumed part of the general DTD, e.g. `no, tt`.

(Direct) TEX markup

As expected, some formatting commands had to be included in order to reproduce the published results.

The approach has been to supply a template (preamble) line for the layout of the contents proper and to handle the header row as an exception to the template, *manually!* Separators between header, body and footer have to be incorporated as well. A translation table is clearly insufficient. The TEX markup, then, would look like:

```
\newdimen\entrywidth%\entrywidth=<default>
\newdimen\tablewidth
\tablewidth=.5\hsize%default
\def\nl{\par\noindent}
\def\ndash{--}
\def\tablerule{\noalign{\hrule}}
\newdimen\digitwidth\setbox0=\hbox{\rm0}
         \digitwidth=\wd0
%?-command for nonsignificant leading
%  zeroes, Knuth p241
\catcode'?=\active
\def?{\kern\digitwidth}
%\btbl %AAP's simple example with direct
%       TeX markup
\entrywidth=2cm
\tablewidth=4\entrywidth
\vbox{\hsize=\tablewidth
\halign{\hbox to\entrywidth{#\hss}\hfil&&
    \hbox to .5\entrywidth{\hss#}\hfil\cr
%preamble line
   \tablerule\noalign{\vskip1ex}
   \omit{\bf  Table AAP} Job Changes:
                         1973--1980
   \hidewidth\cr
   \tablerule\noalign{\vskip1ex}
   \omit           &
   \omit\vtop{\noindent\hsize=\entrywidth
        Gain/Loss\nl
        of Hospitals \nl
        since 1973}&
   \omit\vtop{\noindent\hsize=\entrywidth
        Total No.    \nl
        of CEO       \nl
        Job Changes  \nl
        1973--80}  &
   \omit\vtop{\noindent\hsize=\entrywidth
        Survival     \nl
        Rate of      \nl
        CEO's}     \cr%end header row
\noalign{\vskip.5ex\hrule\vskip.5ex}
%head-body separation
   Texas   &$+$20&---& 22\%\cr
   Maryland&$+$?5&42 & 24\%\cr
\noalign{\vskip1ex}%body-foot separation
   \noalign{Source: David Kinzer,
   ''Turnover Of Hospital Chief Executive
     Officers:
     A Hospital Association Perspective,''
     {\it Hospital and health Services
     Administration\/} May--June 1982.
    }%end \noalign
}%end \halign
}%end \vbox
```

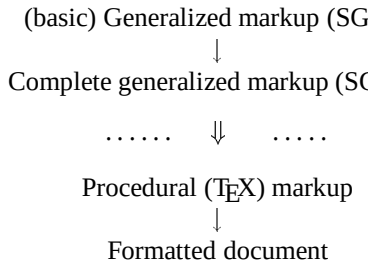
Transformation revisited

Complete SGML markup with procedural TEX markup representation can be achieved via the SGML link mechanism, or by any automatic substitution process (programmable editor, preprocessor). Translation into TEX can be done "direct" or via procedural markup. For the

mathematical examples given in van der Laan, Coleman [1989], this has been done by Grootenhuis [1990]. He has used the SGML link mechanism for "substitution" and did direct coupling to LATEX. The *direct* coupling approach, without procedural (LATEX) markup, has the disadvantage that change of formatter requires (TEX) source adaptation. In order to abstract from any particular format, we have considered procedural TEX markup as an intermediate phase, in van der Laan, Coleman [1990].

SGML⇒TEX using procedural markup

The "generalized markup ⇒ format" process can be characterized by the following four levels,



with the input phase and output (print) phase before and after. The interfacing with procedural markup is illustrated below, with AAP's simplified table as example. This four level process resembles the coupling of higher level programming languages such as PASCAL and ADA to FORTRAN (numerical libraries). For more on the latter, I refer to Einarsson [1988] and other early work of mine [1984].

Note. Of course, one can also work the other way round: start from procedural marked up copy and couple that to SGML.

(Completely tagged) SGML markup, AAP's table

The complete SGML markup version—complete means expansion of short references into tags and addition of omitted (end) tags—of AAP's simple table reads:

```
<tbl>
<no>Table AAP</no>
<tt>Job Changes: 1973&ndash;1980</tt>
<hd>
   <th></th>
   <th>Gain/Loss of Hospitals
       since 1973</th>
   <th>Total No. of CEO Job Changes
       1973&ndash;80</th>
   <th>Survival Rate of CEO's</th>
</hd>
<bdy>
<row><tsb> Texas</tsb>
     <c>20</c><c></c><c>22%</c>
</row>
<row><tsb>Maryland</tsb>
     <c>5</c><c>42</c><c>24%</c>
</row>
</bdy>
<ft>
   <au>David Kinzer</au>
   <atl>Turnover Of Hospital Chief
        Executive Officers: A Hospital
        Association Perspective</atl>
   <src>Hospital and health Services
        Administration</src>
   <dt>May&ndash;June 1982</dt>
</ft>
</tbl>
```

The above tagged table describes contents and structure. The variety of presentations for printing must be catered for with either a TEX format or a LATEX style file.

(Procedural) TEX markup, AAP's simple table

We have limited ourselves to "translating" SGML markup into procedural TEX markup (no LATEX markup). (Mainly: `<name>` into `\bname` and `</name>` into `\ename` ; the tags can be transformed table driven, but in the header row `\nl` commands have to be inserted manually, guided by taste and aesthetics and limited by the value of `\entrywidth`. Note that the data have to be adapted too: insertion of `?` for suppressed 0, and `\` before %).

```
\entrywidth=2cm
\tablewidth=4\entrywidth
\btbl %AAP's simple example with TeX
      %procedural markup
\bno Table AAP\eno
\btt Job Changes: 1973--1980 \ett
\bhd
   \bth\eth
   \bth Net Gain/Loss\nl
        of Hospitals \nl
        since 1973 \eth
   \bth Total No.    \nl
        of CEO       \nl
        Job Changes  \nl
        1973--80 \eth
   \bth Survival     \nl
        Rate of      \nl
        CEO's \eth
\ehd
\btby
\brow\btsb Texas\etsb\bc$+$20\ec
     \bc\ec\bc 22\%\ec
\erow
\brow\btsb Maryland\etsb\bc$+$?5\ec
     \bc 42\ec\bc 24\%\ec
\erow
\etby
\bsrc
   \bau  David Kinzer\eau
   \batl Turnover Of Hospital Chief
   Executive Officers:
   A Hospital Association Perspective\eatl
   \bnme Hospital and health Services
   Administration\enme
   \bdt May--June 1982\edt
\esrc
\etbl
```

Note. We still have to supply the values for entrywidth and tablewidth along with each particular table, once again, manually.

TEX format macros

In order to reproduce AAP's representation the following (format) macros were written

```
%TeX ''format'' for AAP's simple table.
\newdimen\entrywidth%\entrywidth=<default>
\newdimen\tablewidth
\tablewidth=\hsize%default
\def\nl{\par\noindent}
\def\ndash{--}
%? command for nonsignificant
%  leading zeroes, Knuth p241
\catcode`?=\active
\def?{\kern\digitwidth}
%
\def\tablerule{\noalign{\hrule}}
\def\btbl{\bgroup
   \def\bno##1\eno{{\bf##1}}
   \def\btt##1\ett{{##1}\hidewidth\cr}
   \def\bhd{\tablerule\noalign{\vskip1ex}}
   \def\ehd{\cr
       \noalign{\vskip.5ex}\tablerule
       \noalign{\vskip.5ex}}
   \def\bth##1\eth{\vtop{\noindent
             \hsize=\entrywidth ##1}&}
   \def\btby{\noalign{\vskip1ex}}
   \def\etby{\noalign{\vskip1ex}}
   \def\btsb##1\etsb{\hbox to
             \entrywidth{##1\hss}\hfil}
   \def\bc##1\ec{&\hbox to .5
             \entrywidth{\hss ##1}\hfil}
   \def\brow##1\erow{##1\cr}
   \def\bsrc{\noalign\bgroup}
   \def\esrc{%Source information is
             %handled conform AAP's
             %representation
            Source: \gau, ''\gatl,''
            {\it \gnme\/}
                    \gdt.\ \gobi
            \egroup}
   % Next items are ''stored'' via gdefs
   \def\bau##1\eau{\gdef\gau{##1}}
   \def\batl##1\eatl{\gdef\gatl{##1}}
   \def\bnme##1\enme{\gdef\gnme{##1}}
   \def\bdt##1\edt{\gdef\gdt{##1}}
$$\vbox\bgroup\hsize=\tablewidth
    \halign\bgroup &##\cr%preamble line
    \tablerule\noalign{\vskip1ex}
}%end\btbl
\def\etbl{\egroup%\halign
          \egroup$$%\vbox
          \egroup%\btbl
         }%end \etbl
%end AAP simple table format
```

The above listed format macros take care of the final results in print: appropriate separators and good order and format of the 'source' items. The table entries are centered and aligned on the last digit. This required knowledge of the column width. In order to handle the footer suitably the tablewidth had to be known. The chosen approach allows flexible formatting of the footer. Variability of column widths has not been incorporated in the provided macros but can be addressed.

Difficulties with AAP's complex table DTD

Although not the case in the above elaborated example, we experienced difficulties with header rows which contain "halflines." In my opinion, halflines belong to the structure. Confusion arises when the `br` (begin row) and `er` (end row) attributes are used together with halflines. According to the DTD, halflines don't account for a line in the formatted result, in TEX formatting they do, jeopardizing the prescribed br- and er-values.
Note that author etc. information is stored in gdef s in TEX, in order to cope with the difference in sequence order of this items in SGML (AAP's DTD) and TEX (independent) marked up tables.

Graphics

Coupling graphics is not (yet) elaborated, because graphics in SGML is left to other sources. CGM (Computer Graphics Metafile) methodology is the way SGML would like to see graphics incorporated. (Comm. Malcolm Clark, Idle by the thames, TEXline X.) Various graphic sources are interfaced to SGML. The only structuring aspect deals with open space (to (electronically) paste up the illustration) which must not be split over a page break. The difference with mathematics possibly is that formulas are also part of the text while illustrations are more or less separated from the text.

Developments

A survey of the development of SGML is given by Barron [1990].

Usage

Some uses of SGML would be:

- DOD (Automated Technical Order System)
- European Communities (FORmalised EXchange of Electronic Documents; office official publications)
- Publishers (AAP, British Library, KNUB(Elsevier, Kluwer, . . .), . . .)
- Her Majesty's Stationary Office (legal text)
- HP Technical documentation
- Oxford University Press (abridged forms, database applications)
- McGraw Hill Encyclopedia of Science and technology (CD-ROM)
- SGML Users Group (chapters in various countries)
- TEX to SGML assisting tools: XTRAN, Texttagger, Fasttag (communicated by Eric van Herwijnen)

Plans

Some plans for SGML use are being formulated:

- DOD (Computer-aided Acquisition and Logistic Support)
  Object: To produce an integrated system in which information is held electronically, and which interfaces to CAD/CAM systems, electronic publishing systems and databases and those operated by the many defense contractors who supply the department, so that it will be possible to receive, distribute and use technical information in digital form.
- TEI-project, (Text Encoding and Interchange of machine readable texts), Sperberg-McQueen & Burnard [1990].

Local work in progress

Some local SGML work that is being done is:

- Elsvier's experiment, Bleeker [1989] and Poppelier [1990].
- Tabular material examples, van der Laan, Coleman [1990].
- Coupling SGML to LATEX (mathematics) Grootenhuis [1990].

Guidelines for Choosing

As always whether to choose to use SGML, LATEX, or TEX depends upon many factors:

what is the document like,
what tools are already in use,
for whom is it aimed at,
how many authors are involved
is (partial, e.g. bibliographical) reuse also the case,
is future use, different from formatting, in sight?

No structure For documents with little structure, just standard font use, and no page make-up complexity, it does not matter what is used, provided minimal quality is obtained.

Scientific papers For scientific papers with complex mathematical, or physical, as well as tabular structures, TEX/LATEX can best be used, especially when publishers accept TEX marked copy.

Reuse When reuse is the issue, e.g. bibliographic information, or document parts stored in a database, SGML can best be used.

Various authors When various authors with diverse backgrounds and text processing tools, provide copy to a publisher, it is tempting to consider SGML as a uniform language. Suitable tools with proof facilities are necessary in order to stimilate authors to use it. One can also consider TEX which is *de facto* in use for that purpose.

Future (nonformatting) use Abstract from medium, medium neutrality, and use SGML standard.

And remember: "SGML is on our minds, TEX is in our hands."

What to do Next?

What about (digital) sound and (digital) video as part of the structured information?

Hypertext(, SGML(, TEX(, . . .)· · ·)?

While still struggling with DTDs, formatting, printing and coupling problems, it is tempting to ponder once in a while about *Information* Type Definitions.

Conclusion

SGML may benefit from TEX formatting, and TEX may benefit from SGML descriptive markup. From the mathematical and tabular examples it can be seen that there are difficulties in transforming SGML marked up copy into TEX, when purism is strived for. The pragmatic approach with TEX as a generally accepted SGML NOTATION, for (displayed) mathematical and tabular TEX marked up copy, seems practical. Agreed this limits to formatting, but TEX is stable and will serve a life-time. TEX marked up copy can always be converted when needed. This approach has also been mentioned by TEI (Sperberg-McQueen & Burnard, 1990). Warmer mentions also difficulties in writing DTD s for tabular material because of the dichotomy of the tree structure: based on rows or on columns.

Acknowledgements

Most SGML codings are tentative, only the original SGML codings of mathematics have been parsed. Mary and Dean Guenther are kindly acknowledged for their support in molding the article into acceptable size and for changing the text into palatable English.

References

[1] Appelt, W.(1988): Introduction to TEX. Addison-Wesley.

[2] Association of American Publishers (1987): Standard for electronic manuscript preparation and markup. AAP inc. [8]

[3] Association of American Publishers (1989): Author's guide to electronic manuscript preparation and markup. AAP inc. (2^{nd} version.)

[8] Association of American Publishers, 2005 Massachusetts Avenue, NW. Washington, DC 20036, Phone: (202) 232-3335

[4] Association of American Publishers (1989): Reference manual on electronic manuscript preparation and markup. AAP inc. (2^{nd} version.) ISBN 1-55653-084-6.

[5] Association of American Publishers (1989): Markup of tabular material. AAP Inc. (2^{nd} version.) ISBN 1-55653-085-4.

[6] Association of American Publishers (1989): Markup of Mathematical Formulas. AAP Inc. EPSIG. (2^{nd} version.)

[7] Barron, D.(1989): Why use SGML? Electronic publishing, 2,1, 3–24.

[8] Bechtolsheim, S. von (1990): TEX in practice. (4 volumes, 1200p.).

[9] Bleeker, J.(1989): Electronische verzending, bewerking en opslag van wetenschappelijke artikelen. In: SGML de consequenties. De eerste Nederlandse SGML conferentie. SGML User's Group Holland. (Dutch)

[10] Bryan, M.(1988): SGML, an Author's Guide to the Standard Generalized Markup Language. Addison-Wesley.

[11] Buerger, D.(1990): LATEX for scientists and engineers, McGraw-Hill.

[12] Cheswick, B.(1990): A permuted index for TEX and LATEX. CSR 145. AT&T Bell laboratories.

[13] Clark, M.(1988): A note comparing TEX to SGML. SGML User's Group Bulletin, 3, 2, 67–68.

[14] Clark, M. (1989): TEX and/or SGML. Proceedings EuroTEX89. Karlsruhe. (Context sensitivity as a tool for checking input correctness is stressed; an example of how to do this within TEX is given.)

[15] Coombs, J.H., A.H. Renear, S.J. DeRose (1987): Markup systems and the future of scholarly text processing. Comm. ACM, 30, 11, 933–947.

[16] Doob, M.(1989): A gentle introduction to TEX. A manual for selfstudy.

[17] Einarsson, B.(1988): Mixed Language programming. Sowtware-Practice and Experience.

[18] Genussa, P.L.(1987): Document Preparation Method of the United States Aire Force Automated Technical Order System. SGML Users' group. Bulletin 2, 1.

[19] Grootenhuis, J.(1990): Typesetting SGML coded Mathematics. Paper presented at Markup'90. Charleston.

[20] Guittet, C.(1986): FORMEX: une mise en practique des normes internationales. SGML user's group. Bulletin, 1, 2.

[21] Herwijnen, E. van (1988): Electronic submission of Physics articles to publishers. De 1^e Nederlandse SGML conferentie. SGML: De Consequenties. (Also published in:Computer Physics Communications 57 (1989) 244–250: The use of text interchange standards for submitting physics articles to journals.). In the context of this paper the discussion of SGML related to TEX is relevant.)

[22] Herwijnen, E. van (1990): Practical SGML. Kluwer-Academic.

[23] ISO8879 Information Processing—Text and Office Systems—Standard Generalized Markup Language (SGML). 1986–10–15.

[24] ISO/TR9573 Information Processing—SGML support facilities —Techniques for using SGML. 1988–09–12.

[25] Knuth, D.E. (1989): The Errors of TEX. Softw. prac. exp. 19, 7. 607–685.

[26] Laan, C.G. van der (1984): (Graceful) Mixed Language Programming. Argonne National Laboratory Workshop.

[27] Laan, C.G. van der (1990): Typesetting Bridge via TEX. TUGboat, 11#2, 265–276.

[28] Laan, C.G. van der, D.C. Coleman, J.R. Luyten (1989): SGML–(La)TEX. 1. Mathematical Formulas. RC-RUG report 24.

[29] Laan, C.G. van der, D.C. Coleman (to appear): SGML–(La)TEX. 2. Tabular material.

[30] Lamport, L. (1985): LATEX a Document Preparation System. Addison-Wesley.

[31] Poppelier, N.A.F.M.(1990): SGML and TEX in Scientific Publishing. EuroTEX90, Cork.

[32] Seroul, R.(1989): Le petit livre de TEX. InterÉditions. Paris.

[33] MARK-IT (1989): SGML Parser, version 2. Sobemap NV, Place du Champ de Mars 5, Bte 40, 1050 Bruxelles.

[34] Schwarz, N.(1989): Einführung in TEX. Addison-Wesley. (Translated into Dutch and English)

[35] Smith, J.M.(1987): The standard generalized markup language (SGML): Guidelines for editors and publishers. British National Bibliography Research Fund Report 26. ISBN 0-7123-3111-5.

[36] Smith, J.M.(1987): The standard generalized markup language (SGML): Guidelines for authors. British National Bibliography Research Fund Report 27. ISBN 0-7123-3112-3.

[37] SGML User's Group Newsletters. [9]

[9] Editorial address: Pindar Infotek, 2 Grosvenor Road, Wallington, Surrey SM6 0ER, UK.

[38] Sperberg-McQueen, C.M., L. Bernard (1990): ACH-ACL-ALLC. Guidelines for the encoding and interchange of machine readable texts.

[39] Vignaud, D.(1989): L'édition structrée dans les documents, SGML applications à l'édition française. Éditions du Cercle de la Librairie. Paris.

[40] Warmer, J., S. van Egmond (1989): The implementation of the Amsterdam SGML Parser. EP-ODD, 2, 2, 65–90.

[41] Warmer, J. (priv. comm).

[42] Wittbecker, A.(1989): TEX enslaved. Proceedings TUG89. Stanford. (Advantages and disadvantages of TEX-formatter with SGML "front-end" are discussed, related to DEC's VAX Document.)

BIJLAGE FF

SGML and TEX in scientific publishing

N.A.F.M. Poppelier
Elsevier Science Publishers
Physical Sciences and Engineering Division
R&D Department
n.poppelier@elsevier.nl

Abstract

Elsevier Science Publishers has for a few years investigated the possibility of accepting compuscripts, a manuscript in electronic form, created with TEX, LATEX and a few other text processing systems, and converting these to *SGML* form. This paper will discuss the current status of these activities, the reasons for converting compuscripts to *SGML* form, and the various ways in which TEX is used.

Introduction

Until a few years ago the results of scientific research were always published in the following way: the author writes an article using whatever method he prefers. Some authors use TEX, some authors a word processor, others prefer a typewriter, and there are still authors who consider the pen to be the best desktop publishing tool that is available.

When the author is satisfied with his work, he or she submits the manuscript, say, to the editor of a journal, usually in duplicate or triplicate. The editor enters the manuscripts into his administration, and sends a copy to one or more "referees," who review and judge the manuscript. This is one of the places on the long road from manuscript to published paper where the publisher adds something to the paper. The system of peer reviewing, which provides a kind of scientific certification, is one example of "added value." The quality of the printed product and the world-wide distribution of an article in a journal can also be labelled "added value."

The referee(s) can either recommend acceptance of the manuscript for publication, possibly with suggestions for alterations, or rejection of the manuscript. If it is accepted, and after the author has put it into final form, the manuscript is sent to the publisher, where it is adapted to the style of the journal by a technical editor. Then the manuscript is re-typed (typeset) and, in most cases, stored in electronic form. Because of the fact that the material has been re-typed, it may contain errors. These errors are removed by proof-reading, which is a task that can be performed, for example, by the technical editor or the author.

Finally, the paper is published, together with other papers, in the journal and after a while the author sees his article in print.

The time that passes between submission of the manuscript and publication of the article in printed form can vary between a few weeks and a year, or even longer. In some fields of science this period of time is too long and authors sometimes ask publishers why this time cannot be reduced.

Modern Times

The conventional method of publishing, which I have sketched above, can probably not be made any more efficient by using conventional means. However, many authors use their own text processing systems to produce their papers and reports. This text can be transmitted in electronic form. In principle, this makes it possible to

- publish the paper within a shorter length of time;
- extract bibliographic information and abstracts, and store this information in a database.

In the first case we see that, by making use of manuscripts in electronic form, which I will call *compuscripts* in the rest of this paper, that is by making use of modern electronic means, the conventional method of publishing scientific journals *can* be made more efficient. Furthermore, the production of secondary publications, such as abstract journals, can be made more efficient since the abstracts do not have to be re-typed.

An important consequence is that the publisher can now add extra value to the journal or to his entire range of scientific publications, by making the material available in various electronic forms, such as hypertext books, databases on CD-ROM, etc.

However, a publisher can only accept a compuscript if the instructions of the text processing system of the author can, in one way or another, be converted to that of the publisher, which in most cases is a computer-driven phototypesetter. Since a publisher wants to work as effi-

ciently as possible, this conversion must be automatic or nearly automatic. This means that the compuscript must be prepared by the author according to a set of rules. In most cases, of course, a technical editor still has to read a draft printed on paper to mark appropriate changes to the text, the typography or the layout. Especially if the paper contains tables or mathematical formulae, this is an intricate task.

Research at ESP

Elsevier Science Publishers (*ESP*) has for a few years been investigating the possibility of accepting compuscripts created with modern text processing systems. The aim of our present research is to develop a method for manipulating the compuscripts in such a way that the material can be published more efficiently than before, and is also stored for re-use at a later stage.

We take the view that the transition from manuscript-based to compuscript-based publishing must at least preserve the flexibility of the current submission procedures and, where possible, improve upon it. This means, for instance, that the author should not be bothered with compuscript preparation instructions that are significantly different between publishers, or even between the various journals of one publisher. It also means that publishers should be able to accept compuscripts produced by a range of text processing systems, and submitted either by electronic network or on a diskette, via ordinary mail.

SGML

As Figure 1 shows, the Standard Generalized Markup Language (SGML) [1, 2, 3] plays a central role in our approach. SGML promises to become a very important tool, since it allows us to (i) separate the logical structure of the document from its visual structure, i.e. its appearance; (ii) treat all accepted compuscripts, once they have been converted to SGML form, in a uniform working environment; (iii) handle the compuscripts independently of output medium and output format; (iv) store the text, wholly or partially, in a database and use the database contents again at a later stage for various other forms of publications; (v) standardize compuscript input formats between the various scientific publishers, thus helping to preserve the flexibility of submission mentioned above.

It is not our intention to ask authors to submit SGML-coded compuscripts yet. However, we foresee that, as a consequence of the fact that the departments of the US Government as well as major industries have adopted SGML as a standard for document interchange, developers of text processing software will provide their customers with facilities for conversion from their particular text format to SGML. Authors will then be able to produce their compuscripts with the text processing system of their choice and to provide the publisher with an SGML version without additional effort.

Until then, in order to have all material submitted in the form of compuscripts available in SGML-coded form, conversion facilities must be developed. The scheme we would like to realize is indicated in Figure 1.

On the one hand, Elsevier Science Publishers wants to allow authors to use their favorite text processing system. On the other hand we know that there are dozens of text processing systems that are being used by authors in various disciplines.

This means that we have to make a choice. Two factors influence this choice: (i) the text processing systems chosen must correspond to a fair proportion of the total number of authors, and (ii) a compuscript prepared with a certain text processing system must, ideally, be convertible to SGML form. The first point is something that should be investigated separately for every discipline. The second point implies that we must choose text processing systems that, whenever possible, produce texts where the logical structure is, at least partially, indicated. In other words: a Word compuscript based on a style sheet is to be preferred over a plain Word file, and LaTeX is preferred over plain TeX.

TeX Compuscripts

TeX will be one of the text processing systems included in our electronic publishing scheme. Later this year, as a first step towards having a TeX compuscript scheme for all appropriate journals, *ESP* will start accepting LaTeX-coded compuscripts for a few of their physics journals. For various reasons we have chosen LaTeX as the most convenient variety of TeX.

- Since according to Lamport [4], "the primary function of almost all the LaTeX commands . . . [is] to describe the logical structure of [a] document" conversion of a LaTeX-coded compuscript to SGML form is possible.
- The idea of a document style, which is fundamental to LaTeX, enables us to produce camera ready copy for a particular journal by changing only the `\documentstyle` command in the submitted compuscript.
- LaTeX is relatively easy to learn.
- LaTeX is described in a book, so that it can be considered as a *de facto* standard.
- Especially for large review articles and for books, the following LaTeX tools come in very handy
  - cross-referencing with symbolic keys,
  - automatically generated table of contents,
  - index and bibliography tools.

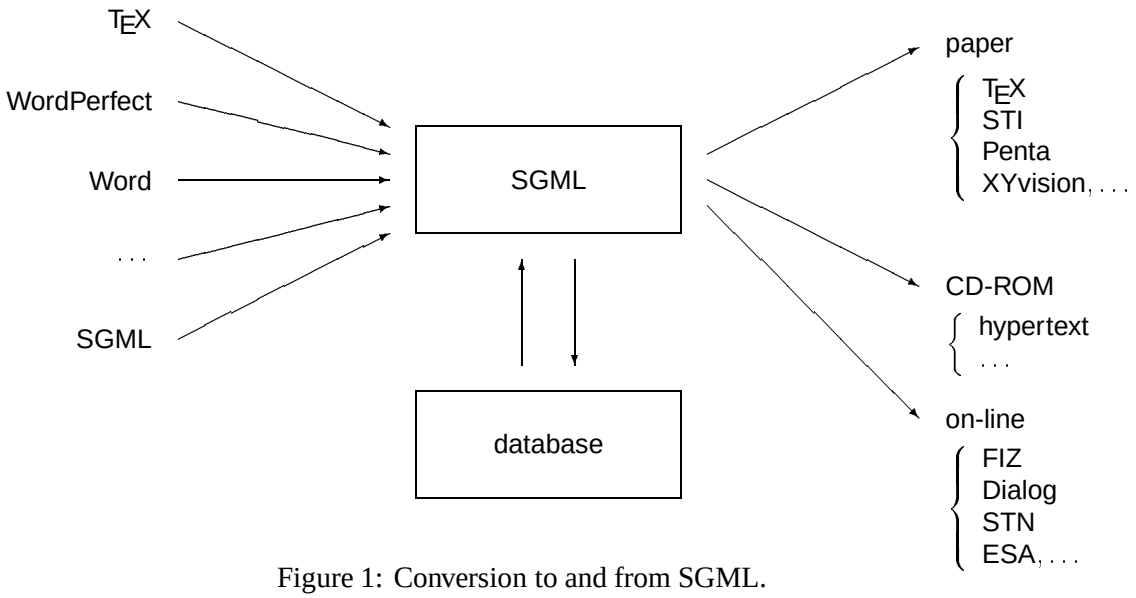


Figure 1: Conversion to and from SGML.

Status Quo

So far we have achieved several things. First of all, we have created LaTeX document styles for four of *ESP*'s scientific journals. These new document styles exactly reproduce the layout of the corresponding journals, which so far are still produced in the conventional way.

We have also been able to develop a set of programs for complete automatic conversion of LaTeX documents with mathematical formulae to SGML form. To be more precise: we convert LaTeX documents with mathematical formulae to a *document type definition* [1, 2] for scientific papers that is based on the one developed by the *Association of American Publishers* (*AAP*).

A document type definition is a description of the logical structure of a certain class of documents, using a method that resembles the specification of the syntax of a programming language. As an example, the document type definition of a novel is represented in a diagrammatic manner in Figure 2.

Conversion of LaTeX-coded tabular material to SGML form, i.e, to the *AAP* document type definition or possibly a different document definition, is currently under investigation. Another automatic conversion that is currently under investigation is the conversion from Word, with an appropriate style sheet, to SGML.

Book Projects with TEX

Recently the Physical Sciences and Engineering Division of *ESP* has published several books that have been written in plain TeX or in LaTeX, and the number of TeX-coded and LaTeX-coded books that we will publish in the near future will probably increase. Our experience is that LaTeX is an excellent tool for these larger projects due to the separation of logical structure and visual structure, the ability to do cross-referencing with symbolic keys and the presence of tools for index and bibliography; these advantages were already mentioned under the section *TEX Compuscripts*.

However, we have also noticed that making a TeX-coded or LaTeX-coded book ready for publication requires more time when the authors do not have to follow a set of instructions, but are free to type things the way they think is best.

Other Uses of TEX

Lots of programs, such as database programs, have output capabilities that are somewhat meager. After all, a database program is meant to store data, not to print it in a sophisticated manner. The fact that TeX is a programmable text formatting system makes it an excellent "print engine" for such programs. As long as the database program is capable of inserting some fairly simple texts in its output – most database programs have report generators that are able to accomplish this – TeX's capabilities as a text formatter can be used to print the output in any desired way.

An example of this approach is symbolic mathematics packages that can compute complicated formulae and generate the TeX codes for formatting them. Another example is provided by the database-publishing activities of the *Excerpta Medica Publishing Group* (*EMPG*), which is part of the Biomedical Division of *ESP*.

The *EMPG* employs a database of article openings, abstracts and citation lists that have been derived from articles in a selected set of biomedical journals. The information in this database is stored in a structured form. With this information the *EMPG* produces abstract journals that are called the *Core Journals in* ⟨ . . . ⟩,

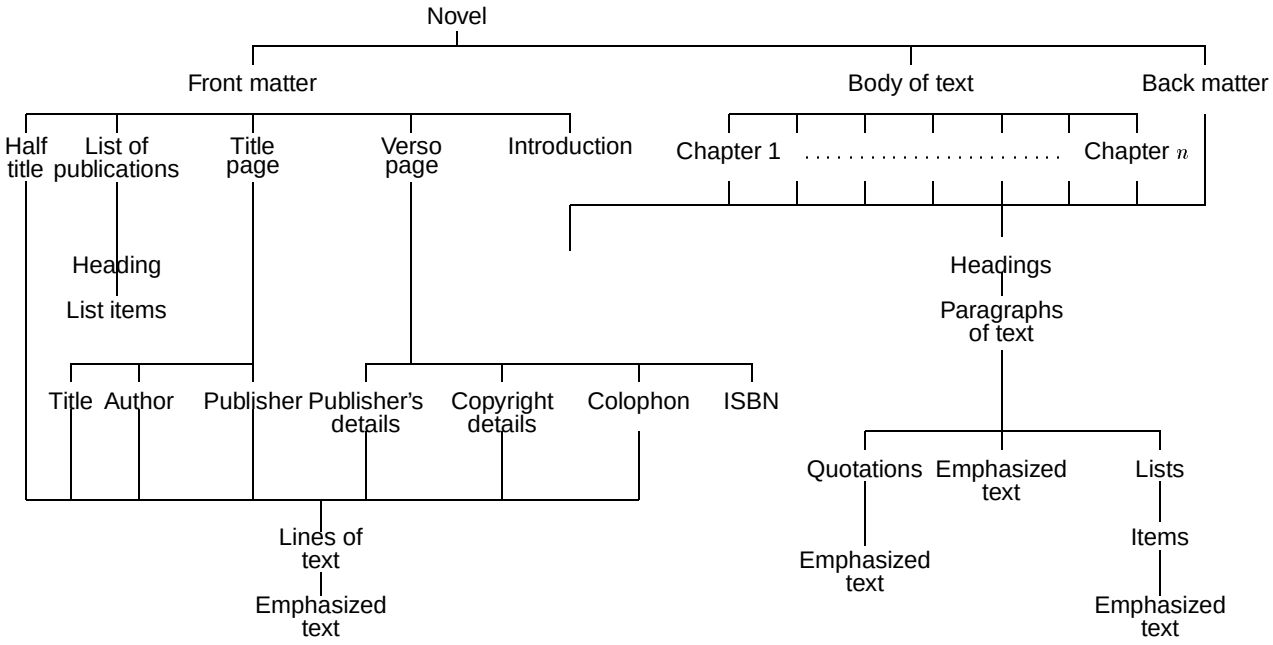


Figure 2: Document type definition of a novel.

where $\langle \ldots \rangle$ is, for instance, *Opthalmology*, *Cardiology* or *Neurology*. The *EMPG* has been experimenting with using LaTeX for the production of *Core Journal*s: the relevant portions of the database are extracted and converted to LaTeX. The LaTeX-coded form of this information can then, in principle, be used to produce the camera ready copy for the journal, using a document style that was developed in-house.

In the near future, starting this year, a similar system will be implemented for the article openings of all journals that are published by *ESP*. The purpose of this project, which is called *CAPCAS*, is to have the article openings, i.e. title, author(s), abstract, keywords and publication history, available in SGML-coded form and to store this information in a database. From this database we can then create secondary publications of various kinds, such as volume indexes, author indexes and abstract journals. Also for this type of database-publishing, TeX is used to format the structured information on paper.

Conclusion

In this paper I have sketched the ideas that *Elsevier Science Publishers* have developed concerning the handling of author prepared manuscripts in electronic form. In this scheme, SGML will play a central role, and TeX, in one or more of its varieties, will also play a part.

Furthermore, I have given a few examples of other ways in which TeX's text formatting capabilities can be put to good use, namely in the fields of book production and database publishing.

Summarizing, I think it's no exaggeration to say that, because of its programmability, TeX is eminently suitable for large-scale text production, both directly as a text processing system, and indirectly as the output component of a database-publishing system. Considering that

- the quality of material typeset in TeX is considered satisfactory even by very demanding users,
- TeX runs on an impressive number of different types of computers, and
- TeX output can be printed on a wide range of output devices,

the number of uses of TeX will undoubtedly increase in the years to come.

Bibliography

[1] International Standard ISO 8879: *Standard Generalized Markup Language (SGML)*. ISO (1986).

[2] Martin Bryan: *SGML, an author's guide to the Standard Generalized Markup Language*. Addison Wesley, (Workingham, 1988)

[3] Eric van Herwijnen: *Practical SGML*. Kluwer Academic, (Dordrecht, 1990).

[4] Leslie Lamport: *LaTeX, a document preparation system*. Addison Wesley, (Reading MA, 1986).

BIJLAGE GG

The future of TEX and METAFONT

Donald E. Knuth

3 october 1990

My work on developing TEX, METAFONT, and Computer Modern has come to an end. I will make no further changes except to correct extremely serious bugs.

I have put these systems into the public domain so that people everywhere can use the ideas freely if they wish. I have also spent thousands of hours trying to ensure that the systems produce essentially identical results on all computers. I strongly believe that an unchanging system has great value, even though it is axiomatic that any complex system can be improved. Therefore I believe that it is unwise to make further "improvements" to the systems called TEX and METAFONT. Let us regard these systems as fixed points, which should give the same results 100 years from now that they produce today.

The current version number for TEX is 3.1, and for METAFONT it is 2.7. If corrections are necessary, the next versions of TEX will be 3.14, then 3.141, then 3.1415, . . . , converging to the ratio of a circle's circumference to its diameter; for METAFONT the sequence will be 2.71, 2.718, . . . , converging to the base of natural logarithms. I intend to be fully responsible for all changes to these systems for the rest of my life. I will periodically study reports of apparent bugs, and I will decide whether changes need to be made. Rewards will be paid to the first finders of any true bugs, at my discretion, but I can no longer afford to double the size of the reward each year. Whenever I have created a new version, I will put it in the official master TEX archive, which currently resides at Stanford University. At the time of my death, it is my intention that the then-current versions of TEX and METAFONT be forever left unchanged, except that the final version numbers to be reported in the "banner" lines of the programs should become

```
TeX, Version $\pi$
```

and

```
METAFONT, Version $e$
```

respectively. From that moment on, all "bugs" will be permanent "features."

As stated on the copyright pages of Volumes B, D, and E, anybody can make use of my programs in whatever way they wish, as long as they do not use the names TEX, METAFONT, or Computer Modern. In particular, any person or group who wants to produce a program superior to mine is free to do so. However, nobody is allowed to call a system TEX or METAFONT unless that system conforms 100% to my own programs, as I have specified in the manuals for the TRIP and TRAP tests. And nobody is allowed to use the names of the Computer Modern fonts in Volume E for any fonts that do not produce identical `tfm` files. This prohibition applies to all people or machines, whether appointed by TUG or by any other organization. I do not intend to delegate the responsibility for maintainance of TEX, METAFONT, or Computer Modern to anybody else, ever.

Of course I do not claim to have found the best solution to every problem. I simply claim that it is a great advantage to have a fixed point as a building block. Improved macro packages can be added on the input side; improved device drivers can be added on the output side. I welcome continued research that will lead to alternative systems that can typeset documents better than TEX is able to do. But the authors of such systems must think of another name.

That is all I ask, after devoting a substantial portion of my life to the creation of these systems and making them available to everybody in the world. I sincerely hope that the members of TUG will help me to enforce these wishes, by putting severe pressure on any person or group who produces any incompatible system and calls it TEX or METAFONT or Computer Modern—no matter how slight the incompatibility might seem.