

Die T_EXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung T_EX e.V.

14. Jahrgang Heft 3/2002 Oktober 2002

3/2002

Impressum

„Die T_EXnische Komödie“ ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit Hilfe folgender Programme erstellt: `pdfTeX`, 3.14159-1.00b-pretest-20020211 (`Web2C 7.3.7x`), `LaTeX2e` (2001/06/01), `Acrobat Reader 4` und `xdvi(k) 22.40k` für die Bildschirmdarstellung. Als Standard-Schriften kamen die Type-1-Fonts CM-Super zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: dante@dante.de

dtk-redaktion@dante.de (Redaktion)

Druck: Konrad Tritsch Print und digitale Medien GmbH
Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohe Stadt

Redaktion: Gerd Neugebauer (verantwortlicher Redakteur)

Luzia Dietsche	Gert Ingold	Volker RW Schaa
Rudolf Herrmann	Rolf Niepraschk	Herbert Voß
Moriz Hoffmann-	Günter Partosch	
Axthelm	Bernd Raichle	

Redaktionsschluss für Heft 4/2002: 23. Oktober 2002

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

diese Ausgabe der Mitgliederzeitschrift steht im Zeichen der „Tipps und Tricks“. In dieser „kleinen“ Form eines Beitrages können viele nützliche Informationen weitergegeben werden. Es muss nicht jedesmal ein seitenschwerer Artikel sein. Es muss auch nicht immer ein neuer Trick sein, der beschrieben wird. Auch eine Beschreibung der Anwendung von existierenden Paketen kann interessant sein. Alle Mitglieder bekommen schließlich mehrmals im Jahr ein paar CD-ROMs zugeschickt – und wer kann da schon alles selbst ausprobieren?

Auch auf diese Weise kann man anderen Lesern weiterhelfen. Ich hoffe, dass sich viele Nachahmer finden und den in dieser Ausgabe beobachtbaren Trend fortsetzen.

Eine andere Beobachtung habe ich auch gemacht, als ich mir die Beiträge in dieser Ausgabe angesehen habe. Es muss nicht immer (\LaTeX) im engeren Sinne sein. Es kann auch sehr interessant sein, die Randgebiete auszuloten, mit denen wir es doch auch immer wieder zu tun haben.

Nicht zuletzt befindet sich in dieser Ausgabe der Mitgliederzeitschrift wieder einmal ein Beitrag zu $\mathcal{N}\mathcal{T}\mathcal{S}$. Endlich können wir einmal sehen, dass $\mathcal{N}\mathcal{T}\mathcal{S}$ nicht nur ein TEX -kompatibler Nachfolger ist, sondern auch, dass es möglich ist, in diesem System interessante Erweiterungen unterzubringen.

So verbleibt es mir nur noch, allen Lesern zu wünschen, dass etwas Passendes in dieser Ausgabe zu finden ist. Und schließlich kann ich auch dieses Editorial nicht zu Ende gehen lassen, ohne alle aufzurufen, sich mit einem kleinen oder größeren Beitrag an der Vereinszeitschrift zu beteiligen.

Mit TEX nischen Grüßen

Ihr Gerd Neugebauer

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

da man immer mit den guten Nachrichten beginnen soll: Die Frühjahrstagung DANTE 2003 wird vom 2.–4. April an der Universität Bremen stattfinden, ausgerichtet von Dr. R. Weibezahn am Institut für Wissenschaftliche Datenverarbeitung. Die Tutorien und Mitgliederversammlung im Herbst 2003 organisiert unser Schriftführer Günter Partosch. Sie finden am 8. und 9. September 2003 im Schloss Rauschholzhausen bei Marburg, der Tagungsstätte der Universität Gießen statt. Hier wird den Teilnehmern Unterkunft und Verpflegung zu günstigen Preisen geboten.

Mittlerweile wurde auch, erstmals in Euro, der Einzug der Mitgliedsbeiträge per Lastschrift durchgeführt. Um zukünftig Rückläufer zu vermeiden, bitten wir, einen Wechsel der Bankverbindung dem Büro mitzuteilen.

Bei Entstehen dieses Grußwortes stand das zweite CD-ROM-Projekt des Jahres an, der CTAN-Abzug. Dieser wird wieder von einem der Unterzeichner (Klaus Höppner) durchgeführt. Ob dies innerhalb des gewünschten Zeitplans stattgefunden hat, sehen Sie daran, ob er dieser Ausgabe beiliegt.

Zu einer Panne kam es bei der Beilage des Druckes unseres Ehrenmitgliedes Prof. Hermann Zapf in „Die T_EXnische Komödie“ Ausgabe 2/2002. Leider wurde der Druck in unserer Druckerei, die auch für Verpackung und Versand zuständig ist, maschinell gefalzt, obwohl Prof. Zapf darum gebeten hatte, dies nicht zu tun, sondern den Sonderdruck locker um Zeitschrift und T_EXLive-CD-ROM zu falten. Wir stehen mit der Druckerei und Prof. Zapf in Kontakt, um für eine fehlerfreie Nachlieferung des Drucks zu sorgen.

Mit freundlichem Gruß

Volker RW Schaa Klaus Höppner
(Vorsitzender) (Stellvertretender Vorsitzender)

TEX-Tagung DANTE 2003 in Bremen – Einladung und Call for Papers

Roland Weibezahn

Die TEX-Tagung DANTE 2003 findet

vom 2. bis 4. April 2003
an der Universität Bremen

statt. Veranstalter sind gemeinsam die Universität Bremen, Institut für Wissenschaftliche Datenverarbeitung und DANTE e.V.

Am Mittwoch sind Tutorien geplant, Donnerstag und Freitag sind für Vorträge und die Mitgliederversammlung von DANTE e.V. vorgesehen.

- o Alle, die einen *Vortrag* oder ein *Tutorium* halten möchten, können dies mit dem Web-Formular „Anmeldung von Beiträgen“

<http://www.dante.de/dante2003/cfp/>

oder per E-Mail an den Organisator

dante2003@dante.de

bis zum

31. Dezember 2002

anmelden. Die Annahme von verspäteten Anmeldungen ist nur unter Vorbehalt möglich.

Zu einem Vortrag oder Tutorium ist ein *extended abstract* einzureichen. Richtlinien dafür sind auf der oben genannten WWW-Seite zu finden.

- o Alle Firmen und Institutionen, die ihre Produkte präsentieren bzw. die Tagung finanziell unterstützen möchten, werden gebeten, sich möglichst frühzeitig an dieselben Adressen zu wenden.
- o Die Homepage der Tagung findet sich unter

<http://www.dante.de/dante2003/>

Hier finden Sie die jeweils aktuelle Fassung des Tagungsprogramms, Hinweise zum Tagungsort, zur Anreise und zu Übernachtungsmöglichkeiten sowie Formulare für die Anmeldung zur Tagung.

- Mit Fragen, Wünschen und Anregungen wenden Sie sich bitte an

Dr. Roland Weibezahn
DANTE 2003
Universität Bremen
Institut für Wissenschaftliche Datenverarbeitung
Postfach 330 440
28334 Bremen
E-Mail: dante2003@dante.de

Wir hoffen, dass möglichst viele $\text{T}_{\text{E}}\text{X}$ -Interessierte unsere Veranstaltung in Bremen besuchen werden, und freuen uns auf einen erfolgreichen Tagungsverlauf.

Bretter, die die Welt bedeuten

Poster im Format DIN A0 mit `a0poster` und `textpos`

Rudolf Herrmann

Die Entstehung eines Posters im Format DIN A0 zur Präsentation auf einer wissenschaftlichen Tagung wird verfolgt. Es werden die \LaTeX -Bordmittel sowie die Pakete `a0poster` und `textpos` eingesetzt. Alternative Vorgehensweisen werden kurz diskutiert.

Ab initio

Wissenschaftler haben des Öfteren die Aufgabe oder auch das Vergnügen, ihre Forschungsergebnisse anderen mitteilen zu müssen oder zu dürfen. Neben den eher biederen Artikeln in Fachzeitschriften (meist Grau in Grau bzw. Schwarz auf Weiß) laden Fachtagungen dazu ein, es auch einmal in Farbe und im Großmaßstab zu versuchen. So etwas stand für meine Frau und mich ebenfalls auf dem Programm; unsere gemeinsamen (älteren) Resultate und ihre neueren Erkenntnisse (mit unverbrauchtem neuen Mitarbeiter) sollten der (hoffentlich) erstaunten Fachwelt zur Kenntnis gebracht werden. Wie schon früher haben wir uns entschlossen, auch diesmal dem \TeX -Leo den Löwenanteil der Arbeit zu übertragen.

Der Artikel verfolgt Schritt für Schritt den Entstehungsprozess und erläutert die verwendeten Pakete. Viele Wege führen zu einem aussagekräftigen Poster; so wurde die Verwendung des externen Programms `poster`, das die PostScript-Ausgabe von \LaTeX bzw. `dvips` nachbearbeitet, bereits einmal beschrieben [5]. Damit ist es grundsätzlich möglich, jedes einseitige \LaTeX -Dokument in ein beliebig skaliertes Poster umzuwandeln. In einem späteren Kapitel wird darauf kurz eingegangen. Bei der Erzeugung dieses einseitigen \LaTeX -Dokumentes sollte man jedoch nach anderen Gesichtspunkten

vorgehen als bei einem normalen Text. Ein Poster ist viel eher eine *Graphik* als eine Buchseite. Daher sind die üblichen Textformatierungs- und Layout-Mechanismen von \LaTeX nur begrenzt einsetzbar. Weshalb wir trotzdem \LaTeX verwenden wollen: Wir schätzen dessen Zeilenumbruchs- und Listenfähigkeiten und einige andere Eigenschaften sehr. Und außerdem sind wir's halt so gewohnt.

Das fertige Poster, das am Ende des Artikels abgebildet ist, ist nicht identisch mit dem, das auf der Tagung gezeigt wurde. Der Grund liegt darin, dass bei einer Verkleinerung von A0 auf die in „Die \TeX nische Komödie“ verwendete Papiergröße A5 einige Details verloren gehen. Ich habe daher im Poster Text weggelassen und stattdessen lieber die eingebundenen Graphiken etwas vergrößert. Das Beispiel sollte deshalb nicht nach wissenschaftlichen Gesichtspunkten betrachtet werden, sondern eher nach der Optik. Da meine Frau die Schrift **Times-Roman** bevorzugt, mir aber die Computer-Modern-Schriften lieber sind, wurde das Original mit dem Paket `times` gesetzt. Dieses Paket war verwendbar, da keine Mathematikfonts benötigt wurden. Generell sollte man lieber `mathptmx` einsetzen, das sowohl für Text wie für Mathematik **Times-Roman** einbindet.

Unser Beispiel wurde hingegen mit dem Paket `ae` gesetzt. Ohne dieses Paket sind die übergroßen Schriftgrößen nicht zugänglich.¹

In medias res

Nachdem wir uns über Inhalt und ungefähres Layout verständigt haben, legen wir nunmehr los. Einen Nachmittag haben wir dafür reserviert. Während die meisten Kollegen an der University of Surrey bedrückt durch die Gänge schleichen, England ist am Vormittag aus der Fußballweltmeisterschaft rausgefliegen, sind wir ziemlich ungestört.

Meine Frau arbeitet an einem recht neuen Rechner mit 1,2 GHz unter Windows XP; als armem Besucher aus Portugal steht mir ein nicht mehr ganz taufrisches Exemplar mit 233 MHz zur Verfügung. Immerhin haben wir darauf Linux installieren können, ohne dass sich jemand beschwert hätte, sodass uns alle Optionen offen stehen. Bevor wir aber daran gehen können, Text und Graphik auf dem Papier zu verteilen, brauchen wir zuerst das Futter für

¹ Ich danke Rolf Niepraschk für die Ersetzung der Bitmap-Fonts durch Type1-Fonts in der PostScript-Datei, die es ermöglicht hat, das Poster ohne größere Qualitätseinbußen hier abzubilden.

den Löwen. Punkt eins ist also die Erstellung der Graphiken. Hier gleich ein weiterer Unterschied zwischen Original und Beispiel: Um eventuelle Verwirrungen britischer Justitiare zu vermeiden, befindet sich im Beispiel an der Stelle des Logos der University of Surrey lediglich das (immerhin VERYHuge geschriebene) Wort „LOGO“.

Cacatum non est pictum

Fast alle Chemiker, die sich unterhalten wollen, benötigen dazu Schreibwerkzeug, da ohne das Zeichnen diverser Formeln die Konversation rasch im Sand verläuft. Strukturformeln sind daher das A und O nahezu aller chemischer Kommunikation. Für L^AT_EX gibt es einige Pakete, die seine Makrofähigkeiten dahingehend erweitern, dass Strukturformeln in den Text eingebunden werden können. Eine Übersicht darüber wurde bereits in „Die T_EXnische Komödie“ vorgestellt [2]. Die Einbindung von Strukturformeln in Fließtext ist jedoch „strongly discouraged“, wie uns die Anleitungen aller chemischen Fachzeitschriften versichern; Formelzeichnungen sind in getrennten Abbildungen unterzubringen. Wie der Autor der Übersicht bevorzugen wir statt dieser Pakete ein separates DOS-Programm zum Formelzeichnen, allerdings ein anderes; es handelt sich um **C-Design**, Version 2.02D aus dem Jahr 1991². Es hat eine Reihe von Nachteilen (insbesondere stört, dass es keine Alternative zu den eingebundenen Nicht-Proportional-Fonts gibt), hat sich insgesamt jedoch bewährt. Für Windows gibt es eine ganze Palette von Zeichenprogrammen, die passabel ihren Dienst tun, für uns aber wegen zu vieler schlecht änderbarer Vorgaben und allgemeiner „Featuritis“ nicht zur Debatte stehen. Für Linux ist mir leider nichts wirklich Brauchbares bekannt; sollte ein GPL-Programmierer sich einmal an so etwas heranmachen, fände er sicherlich begeisterte Abnehmer für sein Produkt.

Meine Frau als geübte Mäusedompteurin mit Designerblick erstellt die Formelzeichnungen mit atemberaubender Geschwindigkeit und in ansprechendem Aussehen. Als Ausgabeformat verwendet **C-Design** ein sehr übersichtlich strukturiertes PostScript, das es erlaubt, auch manuelle Änderungen in der Datei durchzuführen, etwa Teile der Formel einzufärben, wie es in der Graphik oben links geschah, um im Schwarz-Weiß-Druck noch durch unterschiedliche Graustufen erkennbar, Gemeinsamkeiten zwischen unseren Produkten und dem Taxolmolekül hervorzuheben.

²Die DOS-Version ist nicht mehr offiziell erhältlich; es wird nur noch **C-Design** für Windows angeboten (<http://www.ch.tum.de/oc1/EFontain/C-Design/>).

Fiat lux

Sind Formelzeichnungen *symbolische* Repräsentationen von Molekülen, die die Logik des Betrachters ansprechen sollen, so handelt es sich bei Strukturbildern um *realitätsnahe* Abbildungen, die den räumlichen Aufbau von Molekülen möglichst genau wiedergeben sollen, in Analogie zu einer Fotografie. Die üblichen Quellen für solche Strukturbilder sind Röntgenstrukturanalysen und quantenmechanische Berechnungen. Für unser Poster haben wir die Ergebnisse zweier solcher Rechnungen ausgewählt. Sie liegen in Form von xyz-Koordinaten vor, die ein Programm wie PLATON [7] perspektivisch darstellt. Damit drehe ich die Moleküle in günstige Blickwinkel und speichere sie als Input-Dateien für den Raytracer POV-Ray [8] ab. Nach ein paar Änderungen in dieser Datei (so gefällt mir die voreingestellte Hintergrundfarbe nicht, und außerdem ersetze ich die langweilige Farbe Schwarz der Kohlenstoffatome durch eine Granittextur) kann POV-Ray die Datei rendern. Die voreingestellte Beleuchtung ist meist gut auf die Darstellung von Molekülen abgestimmt und unterstützt den dreidimensionalen Eindruck. Das übliche Format des POV-Ray-Outputs ist bei älteren Versionen TGA, bei neueren PNG. Beide können leicht mit einem Programm wie ImageMagick [1] in PostScript überführt werden. Damit haben wir das Material zusammen, aus dem unser Poster aufgebaut werden soll.

Nunc est positionandum

Um unsere Layout-Vorstellungen auf das Papier zu bringen, müssen wir in der Lage sein, Textblöcke und Graphiken an genau definierten Stellen zu positionieren. Im Prinzip hat L^AT_EX mit dem Konzept der `minipage` bereits ein Instrument dafür, das jedoch nicht einfach zu handhaben ist: Änderungen in einer Minipage können zu Positionsänderungen anderer Minipages führen, die nur mit viel Erfahrung und ständigem Üben vorhergesehen und ausgeglichen werden können. Ein sehr präzises Positionieren von Layout-Elementen ist mit dem Paket `textpos` von Norman Gray möglich [3]. Es setzt voraus, dass das Paket `everyshi` von Martin Schröder [6] installiert ist.

Zuerst wird das Paket `textpos` in der Präambel eingebunden:

```
\usepackage[absolute]{textpos}
```

Die Option `absolute` (eine nicht ganz glücklich gewählte Bezeichnung) bewirkt, dass sämtliche Positionierungsbefehle relativ zu einem frei wählbaren

Ursprung auf der Seite wirken. Die Voreinstellung ist allerdings **relative**, d. h. Positionierungen erfolgen relativ zum „current point“ auf der Seite. Um Poster zu erstellen, ist dies aber nicht empfehlenswert. Wir bleiben also bei **absolute** und legen den Ursprung, von der linken oberen Ecke des Papiers an gerechnet, auf

```
\textblockorigin{4cm}{2cm}
```

fest. Schließlich definieren wir noch die horizontalen und vertikalen Einheiten, in denen wir die Breite eines Textblocks und dessen linken oberen Anfangspunkt angeben:

```
\setlength{\TPHorizModule}{30mm}
\setlength{\TPVertModule}{40mm}
```

Damit können wir nun in das Dokument gehen und unsere Layout-Elemente, wie Text und Graphik, nach Belieben auf der (recht großen) DIN-A0-Seite anordnen. Dies geschieht mit der Sequenz:

```
\begin{textblock}{breite}(xpos,ypos)
```

Text und/oder Graphik

```
\end{textblock}
```

Als Einheit von `breite` und `xpos` gilt dabei `TPHorizModule`, in unserem Fall also 30 mm, und für `ypos` `TPVertModule`, hier 40 mm. Alternativ dazu kann auch, unter Verwendung von `textblock*`, die Angabe direkt in Längeneinheiten (z. B. mm) erfolgen. Die angebene Position `xpos,ypos` bezieht sich auf die linke obere Ecke des Textblocks. Ähnlich wie bei einer Minipage ist die Höhe eines Textblocks nicht vorgegeben, sondern richtet sich nach der vertikalen Ausdehnung des Textes oder der Graphik. Anders als Minipages können sich Textblöcke sehr wohl überlappen; dies werden wir später noch ausnutzen.

Die Überschrift des Posters setzt sich aus einer Zusammenballung von Reizwörtern (für Chemiker) zusammen und soll sämtliche Betrachter verlocken, sich alles genau anzusehen und sich später der Autoren zu erinnern – hoffentlich anerkennend. Sie ist daher in der Schriftgröße `VeryHuge` gesetzt (über Größenwahn mehr im nächsten Abschnitt). Das sieht dann so aus:

```
\begin{textblock}{28}(0,0)
  \begin{center}
    {\VeryHuge
      \textsc{Platinum-catalysed Cascade Reactions and}\
      their Application in the Synthesis of}\
      Taxol-like Compounds.}\par}
  \end{center}
\end{textblock}
```

Mit dem manuellen Zeilenumbruch haben wir den Zeilenabstand auf ein unserem Geschmack entsprechendes Maß erhöht. Die Breite des Textblocks beträgt dabei $28 \cdot 30 \text{ mm} = 84 \text{ cm}$; sein Ursprung liegt am `textblockorigin`. Da der Text aber kürzer als die Textblockbreite ist und innerhalb des Textblocks zentriert wird, erreicht er den linken und rechten Papierrand nicht. Der Textblock des Logos überschneidet dabei teilweise den der Überschrift, was aber keine Probleme bereitet, da sich die Texte nicht überlagern (man könnte solche Überlagerungen sogar als stilistisches Mittel einsetzen).

```
\begin{textblock}{6}(0,2.2)
  \begin{center}
    {\VERYHuge LOGO}
  \end{center}
\end{textblock}
```

Dieser Textblock ist also $6 \cdot 30 \text{ mm} = 18 \text{ cm}$ breit, und seine linke obere Ecke liegt an den Koordinaten $x_{\text{pos}} = 4 \text{ cm}$, $y_{\text{pos}} = 2 \text{ cm} + 2 \cdot 40 \text{ mm} = 10,8 \text{ cm}$. Möchte man die Begrenzung der Textblöcke sehen, so lässt sich dies mit

```
\usepackage[absolute,showboxes]{textpos}
```

erreichen; Textblöcke werden dann mit einem Rahmen markiert. Es ist auch möglich, ein Gitternetz über die Seite zu legen; wir haben aber auf solche Maßnahmen verzichtet.

Selbstverständlich können in einem Textblock (fast) alle Stilmöglichkeiten von \LaTeX genutzt werden, etwa Tabellen und Listen. Wie ersichtlich, haben wir mehrmals von der `itemize`-Umgebung Gebrauch gemacht. Die Einbindung von Graphiken erfolgt ebenfalls wie gewohnt, wenn das Paket `graphicx` geladen ist. Die rechte untere Abbildung haben wir folgendermaßen eingebunden:

```
\begin{textblock}{8.5}(18,20.5)
```

```

\begin{center}
  \includegraphics[width=22cm]{prod.ps}

  \vspace{1cm}

  {\LARGE \textbf{Product}}
\end{center}
\end{textblock}

```

Obwohl ich die Graphik `prod.ps` lediglich mit 640×480 Pixel gerendert habe, ergibt sich selbst auf dem DIN-A0-Ausdruck bei einer Breite von 22 cm noch keine markante Unschärfe, wenn man das Poster mit normalem Konferenzabstand ($d_k \geq 1,0$ m) betrachtet. Wer es auch noch mit der Nase auf dem Poster scharf sehen möchte, muss mehr an Render-Zeit und mindestens eine zusätzliche Festplatte investieren.

Maximum es!

Bis jetzt haben wir lediglich Textblöcke auf einem fiktiven Papier von DIN-A0-Größe positioniert. Nun müssen wir \LaTeX beibringen, diese Papiergröße auch wirklich zu verwenden. Dazu dient die Klasse `a0poster` von Gerlinde Kettl und Andreas Weiser [4]. Im Unterschied zu `textpos`, das ständig Ergänzungen und Verbesserungen erfährt, hat sich `a0poster` seit 1997 nicht mehr verändert. Die Datei `a0poster.cls` enthält alle grundlegenden Definitionen, während sich `a0size.sty` um die Fonts kümmert.

Für ein Wahnsinns-Poster braucht man auch Wahnsinns-Schriftgrößen. Die Klasse kleckert wirklich nicht und definiert insgesamt 13 Schriftgrößen, von `tiny` (12 pt) über `normalsize` (25 pt) bis `VERYHuge` (107 pt). Die kleinste in unserem Poster verwendete Schriftgröße ist übrigens `LARGE` (43 pt) ...

Wie funktioniert nun `a0poster`? Erst einmal werden Fonts, Papiergröße, Textbreite und -höhe, Ränder usw. an Übergrößen angepasst. Als Optionen stehen dabei Größen von DIN A3 bis DIN A0 zur Verfügung. So würde man z. B. für den Ausdruck auf einem A3-Drucker (der im Querformat drucken soll) Folgendes in die Präambel schreiben:

```
\documentclass[a3]{a0poster}
```

Wir aber wollen auf DIN A0 drucken. Der am Häufigsten eingesetzte A0-Drucker, der HP Designjet, kann aber noch Papier bedrucken, das etwas

größer als DIN A0 ist. Um diese letzte Reserve ausnützen zu können, gibt es neben der Option `a0` auch noch die Option `a0b` (dies ist sogar die Voreinstellung). Nun wäre es aber praktisch, wenn man einen Probeausdruck des Posters auf einem gängigen A4-Drucker machen könnte, da bei den meisten Nutzern bereits drei Probeausdrucke im A0-Format den Verzicht auf den nächsten Wochenendausflug bedeuten könnten. Dafür ist die Option `draft` gedacht. Wir haben also die Klasse so geladen:

```
\documentclass[portrait,draft]{a0poster}
```

Neben der normalen Erzeugung einer `.dvi`-Datei produziert `a0poster` dabei noch eine Datei namens `a0header.ps`. Diese wird von `dvips` gelesen, wenn es an die Erzeugung der PostScript-Datei geht. In dieser Datei wird dem PostScript-Drucker über den PostScript-Befehl `setpagedevice` mitgeteilt, auf welchem Papier er drucken soll (A0 im Normalfall, A4 bei `draft`), und außerdem wird in letzterem Fall das Dokument verkleinert, um auf eine DIN-A4-Seite zu passen. Dies funktioniert auf jedem PostScript-fähigen Laserdrucker anstandslos, aber nicht mit `ghostview`. Auf dem Bildschirm erscheint das Poster daher immer in der (etwas unübersichtlichen) Originalgröße. Dabei kommt es darauf an, welche Version von `ghostview` auf welchem Betriebssystem man verwendet, ob die Verkleinerung auf Bildschirmgröße gelingt oder nicht. Alternativ kann man daher die PostScript-Datei mittels `ps2pdf`³ in PDF umwandeln und mit dem Acrobat Reader betrachten, der in jeder uns bekannten Version Verkleinerungen zulässt.

Ceterum censeo

Gegen 21 Uhr wäre für mich das Poster soweit zufriedenstellend; meine Frau hingegen stellt mit kritischem Designer-Blick fest, dass es farblich noch nicht ausgewogen ist. Die beiden Formelschemata unter der Überschrift „Two Proposed Reaction Mechanisms“ sollten noch farbig unterlegt werden, „mit pastelligem Grün und Blau“, um ein Gegengewicht zu den Strukturbildern mit blauem Hintergrund am rechten Rand zu bilden. Als Erstes erzeuge ich eine Datei namens `blau.ps` mit folgendem Inhalt:

```
%!  
%%BoundingBox: 0 0 720 1200  
0.8 0.8 1.0 setrgbcolor
```

³ Eine neuere Version von `ghostscript` (≥ 6.0) ist dazu nötig.

```
newpath 0 0 moveto 720 0 lineto 720 1200 lineto
0 1200 lineto closepath fill
showpage
```

Für einen PostScript-Interpreter ist das ein Rechteck, das mit einem „pastelligen Blau“ ausgefüllt ist. Die Farbdefinition verwendet die Grundfarben Rot, Grün und Blau (rgb). Der Pastellton wird dadurch erreicht, dass dem reinen Blau (1.0) ein relativ großer Anteil von Rot und Grün (0.8) beigemischt wird. Analog erhalte ich ein „pastelliges Grün“ mit dem Befehl:

```
0.8 1.0 0.8 setrgbcolor
```

Der Textblock mit dem linken Formelschema ist folgendermaßen aufgebaut:

```
\begin{textblock}{10}(0,14)
  \begin{center}
    \includegraphics[width=21cm]{graphik1.ps}
  \end{center}
\end{textblock}
```

Dieser Block wird nun einfach kopiert und in der Kopie die Zeile mit der eingebundenen Graphik durch

```
\includegraphics[width=30cm,height=47cm]{gruen.ps}
```

ersetzt. Analog verfare ich beim zweiten Textblock mit `blau.ps`. Ganz genau so habe ich auch das LOGO blau unterlegt. Die farbigen Rechtecke sind als PostScript-Graphiken beliebig skalierbar. Die geeigneten Werte für die Höhe und Breite findet man durch etwas Augenmaß und Probieren relativ schnell heraus.

Aber Vorsicht: PostScript-Objekte sind undurchsichtig! Sollte also zuerst das Formelschema gezeichnet werden und erst dann das farbige Rechteck, wird das Schema völlig von der Farbe zugleleistert. Das lässt sich aber recht einfach vermeiden: \LaTeX arbeitet die Textblöcke nach dem Prinzip „Die Ersten werden die Letzten sein“ in umgekehrter Reihenfolge ihres Auftretens in der `.tex`-Datei ab. Also sollte man alle Textblöcke, die flächendeckende Objekte enthalten, wie die genannten farbigen Rechtecke oder Bitmap-Graphiken wie die Strukturbilder, ganz an das Ende (unmittelbar vor `\end{document}`) verschieben.

Audiatu et altera pars

Beim Korrekturlesen dieses Beitrags hat Rolf Niepraschk zwei Alternativen vorgeschlagen, die ich hier kurz vorstellen möchte. Zum Einen gefällt ihm die Klasse `a0poster` nicht, statt dessen sollte lieber das schon erwähnte Programm `poster` [5] eingesetzt werden. Dieses hätte in der Tat den Vorteil, dass jede beliebige andere Klasse verwendet werden kann, die möglicherweise besser an die Eigenschaften des jeweiligen Posters angepasst ist. Außerdem vermeidet man die Schwierigkeiten, die `a0poster` mit den brandneuen `cm-super`-Schriften hat, wenn es um die Riesen-Schriftgrößen geht. Zusätzlich produziert \LaTeX bei den üblichen Schriftgrößen ausgewogenere Zeilenabstände, weshalb auf manuellen Zeilenumbruch in Überschriften verzichtet werden kann. Aus diesen Gründen werden wir sehr wahrscheinlich bei zukünftigen Postern das Programm `poster` zumindest ausprobieren.

Der andere Punkt betrifft das Verfahren, einzelne Teile des Posters farbig zu unterlegen. Statt zu diesem Zweck PostScript-Graphiken einzubinden, verwendet Rolf lieber das Paket `color`, das z. B. folgenden Mechanismus zur Hintergrundgestaltung eines Textblocks ermöglicht:

```
\begin{textblock*}{80mm}(70mm,120mm)%
  \noindent\colorbox[rgb]{0,1,0}{%
    \begin{minipage}[c][50mm][c]{\linewidth - 2\fbboxsep}%
      \centering\Huge\textbf{LOGO}%
    \end{minipage}}%
\end{textblock*}
```

Für dieses Beispiel muss natürlich zusätzlich das Paket `calc` geladen werden. Man erhält damit eine zentrierte Schrift LOGO, die leuchtgrün hinterlegt ist.

Quid novi?

Nunmehr ist der Zeitpunkt gekommen, die Option `draft` aus dem Klassenaufruf zu entfernen und das Poster mittels `dvips` in seiner endgültigen (digitalen) Form zu erzeugen. Ein schnelles `ps2pdf`, das Resultat auf eine Diskette gespeichert, und dann (22 Uhr) heim zum Abendessen (zur Feier des Tages koche ich *Pullus laseratus*, Huhn nach einem Rezept von Apicius, um Christi Geburt, das wir gegen 24 Uhr verspeisen).

PLATINUM-CATALYSED CASCADE REACTIONS AND THEIR APPLICATION IN THE SYNTHESIS OF TAXOL-LIKE COMPOUNDS.

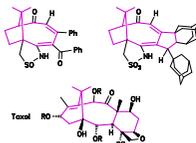
LOGO

Adrian Mahamoodally, Rudolf Herrmann and Gabriele Wagner*

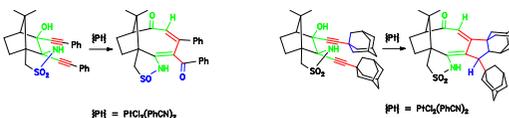
Department of Chemistry, University of Surrey, Guildford GU2 7XH, Surrey (UK).

Aim of the Study

- Development of new cascade reactions for efficient and atom economic synthesis of complex structures
- Understanding these reactions by quantum chemical calculations
- Example of an application: synthesis of Taxol-like compounds

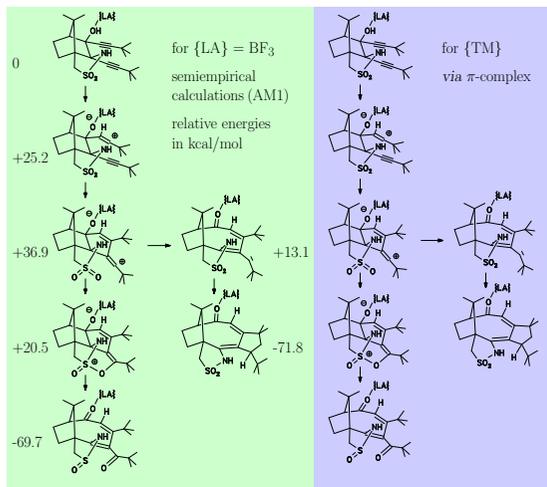


Products of the Cascade Reactions



- cyclisation of the alkynes
 - ring enlargement
 - stereoselective reduction of the sulfonamide
 - only product, no intermediate observed
- reaction is highly selective, but product depends in a surprising way on the alkyne substituents
- cyclisation of the alkynes
 - ring enlargement
 - annulation of a five-membered ring via CH-activation
 - only product, no intermediate observed

Two Proposed Reaction Mechanisms



Calculated Structures



Starting Diyne



Product

References

- [1] G. Wagner, R. Herrmann, A. Schier, *J. Chem. Soc., Perkin Trans.1*, 1997, 701.
- [2] M. F. N. N. Carvalho, A. J. L. Pombreiro, G. Wagner, B. Pedersen, R. Herrmann, *Z. Naturforsch.*, 1999, **54b**, 725.
- [3] A. Mahamoodally, R. Herrmann, G. Wagner, manuscript in preparation.

PDF erfreut sich bei Druckereien steigender Beliebtheit, da man mit ihm nicht viel falsch machen kann; auch die DAD⁴ sollte es noch schaffen, den Inhalt auf Papier zu bringen, so es denn die Kunden bereits druckreif anliefern. In Deutschland und Portugal haben wir früher immer die PostScript-Dateien direkt drucken lassen; als Neu-Engländerin mit noch sehr begrenzter Britannien-Erfahrung will meine Frau aber auf Nummer Sicher gehen und bevorzugt daher PDF. Am nächsten Tag bringt sie die Diskette zum Druckservice der University of Surrey und kann einen Tag später, gegen den stolzen Betrag von £ 72, ein farbenprächtiges Hochglanz-Poster in Empfang nehmen. Auf die angebotene Laminierung (Aufpreis £ 28) hat sie dankend verzichtet. Zu dem Zeitpunkt, zu dem ich (wieder in Portugal) diese Zeilen schreibe, führt sie das Original gerade auf der Tagung in Southampton vor. Wer es gerne in Farbe und A0-Größe sehen möchte statt Grau in Grau bzw. Schwarz auf Weiß und A5 in „Die \TeX nische Komödie“, sollte einmal einen Besuch in Guildford machen. Dort wird es im Department of Chemistry nach Ende der Tagung in einem Schaukasten ausgehängt.⁵

Literatur

- [1] John Cristy: *ImageMagick*; 2000; <http://www.wizards.dupont.com/cristy/ImageMagick.html>.
- [2] Michael Gohdes: *Chemie und \LaTeX – ein schönes Paar?*; *Die \TeX nische Komödie*; 2/01, S. 7–19; 2001.
- [3] Norman Gray: *Textpos – absolute positioning of text on the page*; 2002; CTAN: `tex-archive/macros/latex/contrib/supported/textpos/`.
- [4] Gerlinde Kettl und Andreas Weiser: *a0poster*; 1997; CTAN: `tex-archive/macros/latex/contrib/supported/a0poster/`.
- [5] Rolf Niepraschk: *Poster – leicht gemacht*; *Die \TeX nische Komödie*; 4/00, S. 40–47; 2000.
- [6] Martin Schröder: *EveryShi*; 2001; CTAN: `tex-archive/macros/latex/contrib/supported/ms/`.

⁴Dümmste Anzunehmende Druckerei

⁵Auf besagter Tagung konnte meine Frau übrigens den „best poster award“ einstreichen, bestehend aus einer Flasche „Wickham Fume, Special Release, Estate Bottled, 2000“, und einem Jahresabonnement der *Chemical Communications*.

- [7] A.L. Spek: *PLATON*; *Acta Cryst. A*; 46, S. C-34; 1990; <http://www.cryst.chem.uu.nl/platon>.
- [8] The POV-Ray Team: *Persistence of Vision Ray Tracer*; 2000; <http://www.povray.org>.

Extending $\mathcal{N}\mathcal{T}\mathcal{S}$ (by $\mathcal{X}\mathcal{T}\mathcal{S}$)

Karel Skoupý

The article evaluates the experience gained from extending $\mathcal{N}\mathcal{T}\mathcal{S}$. Two simple extensions are mentioned and a more complicated one is studied in more detail. It involves an abstract model of line-breaking and its application to the *paragraph-breaking into a fixed orthogonal polygon shape*. It shows that extensions of $\mathcal{N}\mathcal{T}\mathcal{S}$ are feasible and discusses the different aspects of making extensions and/or changes.

Introduction

One of the main goals of the $\mathcal{N}\mathcal{T}\mathcal{S}$ project was to provide a framework for experiments with new $\mathbb{T}\mathbb{E}\mathbb{X}$ features and extensions. Making any changes in the standard (Pascal WEB-based) implementation of $\mathbb{T}\mathbb{E}\mathbb{X}$ turned out to be very difficult and unproductive. The $\mathcal{N}\mathcal{T}\mathcal{S}$ as a modular reimplementa-tion of $\mathbb{T}\mathbb{E}\mathbb{X}$ was expected to make the desired changes and extension much easier. Now, that the reimplementa-tion is (almost) finished for more than a year, did we gain any experience with real modifications and extensions? Can the expectations of easier experimenting be met?

Simple extensions

The first two simple extensions appeared around the time of the Euro $\mathbb{T}\mathbb{E}\mathbb{X}$ 2001 conference in Kerkrade. One of them resulted in the introduction of a triple of new primitives: `\vishbox`, `\visvbox`, and `\visvtop`, added by the author. They work exactly like the original primitives without the `vis` prefix but in addition they draw a frame around the corresponding box in

the DVI output. The work was inspired by some features of the package for visual debugging written by Hans Hagen. The difference was that the new primitives in $\mathcal{N}\mathcal{T}\mathcal{S}$ were robust, that means even applying `\lastbox` and other tricks worked as expected. The main reason for trying this extension was to provide some simple example for the workshop on extending $\mathcal{N}\mathcal{T}\mathcal{S}$. The extension was actually so small and local that it was probably not a convincing enough proof of extendibility.

The other extension was the *pluggable file locator architecture* by Simon Peping (see [10]). The “official” (and very simple) way of finding files in $\mathcal{N}\mathcal{T}\mathcal{S}$ is by calling the `kpsewhich` external program. Simon provided an alternative implementation which uses the `kpathsea` library directly via Java Native Interface (JNI) and he changed some $\mathcal{N}\mathcal{T}\mathcal{S}$ code to allow for dynamic configuration of alternative file locators. Although this experiment had a number of interesting aspects (factory design pattern, using JNI, exploring Java networking features) it did not really touch any of $\mathcal{N}\mathcal{T}\mathcal{S}$ “internals”; it rather used the fact that $\mathcal{N}\mathcal{T}\mathcal{S}$ was written in Java than its extendibility.

A real extension

The first *real* (in the author’s opinion) extension of $\mathcal{N}\mathcal{T}\mathcal{S}$ was the *paragraph-breaking into a fixed orthogonal polygon shape* presented at Euro $\mathcal{T}\mathcal{E}\mathcal{X}$ 2002 in Bachotek/Poland. This extension was an attempt to provide an alternative to the rather limited techniques of breaking paragraphs into non-rectangular shapes in $\mathcal{T}\mathcal{E}\mathcal{X}$. Let us describe the problem and its solution in more detail.

Limitations of $\mathcal{T}\mathcal{E}\mathcal{X}$ paragraph shaping

$\mathcal{T}\mathcal{E}\mathcal{X}$ provides a simple mechanism for changing the line width during the line-breaking process by means of `\hangindent`, `\hangafter`, and `\parshape` primitives. The pair of `\hang*` primitives takes care of just one typical case and can always be expressed by `\parshape`.

The `\parshape` specification consists of indents and widths of individual lines of the paragraph in question. By this mechanism, it is possible to achieve arbitrary paragraph shape (but without holes). This kind of specification in terms of lines is however quite unsatisfactory for layouts specified in fixed dimensions (like the cutout for a picture). The problem is that it is not known in advance what will be the vertical size of the individual lines after line-breaking. In simple texts the baselines have constant distances so the

height of a group of lines can be calculated. But as soon as a line contains some irregular object (for example a mathematical formula) the assumption about a constant baseline distance is no longer true.

The adjustment of paragraph shape is traditionally done by humans. It is possible to estimate the number and height of lines, to join more paragraphs with hard breaks if necessary, and to check the result optically. If the desired geometric shape is not matched, then subsequent fine tuning of `\parshape` specification and other parameters is necessary. After modifications of the text, this process must be repeated.

To a certain extent the process described above can be automated. The \LaTeX packages `picinpar`, `floatfig`, and `wrapfig` and the system Con \TeX t provide features for wrapping text around rectangular figures. Both use the `\parshape` or `\hangindent` primitive internally. Algorithmic guessing of the number of lines and subsequent checking of the result is quite a tricky business in the \TeX macro-language and these techniques are not really reliable in difficult cases.

The extension discussed below shows that handling of text shapes given by dimensions is more natural, efficient, and reliable, if it occurs in the place where the decision about line-break is taken rather than trying to make preliminary estimations or subsequent checks. The best place is inside the line-breaking algorithm.

Features available in Adobe InDesign

It seems that commercial desktop publishing tools can traditionally handle the various text shapes quite well. It might be fairly inspiring to look at features provided by Adobe InDesign 2.0¹ which is considered to be the leading application in the field.

All InDesign text resides inside containers called *frames* of which there are two types. The first type is delimited by a closed curve and the contained text is composed into horizontal lines. It can also contain holes of arbitrary shape with text wrapping around them. The second type consists of a single line of text flowing along a general curve. Both kinds have in and out text ports and can be connected to each other to form a chain. The text flows through the chain and adjusts the location automatically when the frame

¹ *Adobe* and *InDesign* are trademarks of Adobe Systems Incorporated in the United States and/or other countries.

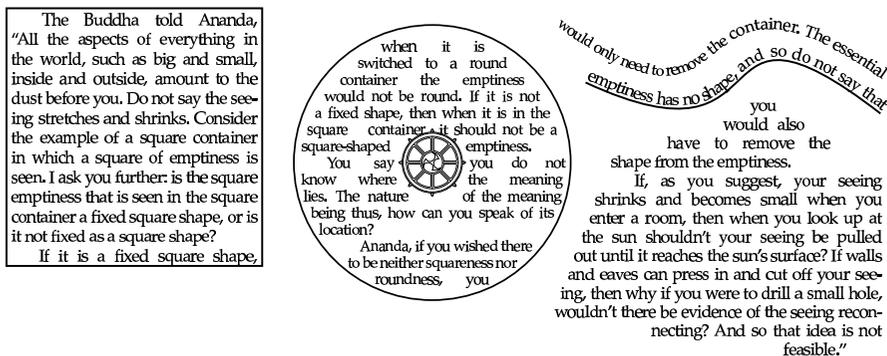


Figure 1: Various text shapes created in Adobe InDesign.

(The text of this illustration was adopted from: Shurangama Sutra [Da Fo Ding Shou Leng Yan Jing], (Taisho Tripitaka, No. 945), URL: <http://hjem.get2net.dk/civet-cat/zen-writings/shurangama-sutra.htm>)

sizes change. An application of the described features is demonstrated in Figure 1.

Most of the frames in a typical InDesign document are used for pages or columns and they are rectangular without holes. Text formatting in frames of non-rectangular shape is somewhat limited. The difference is that the text cannot be justified vertically and is always aligned to the top of the frame. The top of a general shape can be delimited by a non-horizontal or curved line. In such cases, it is interesting to observe how the vertical position of the first line is determined. It turns out that InDesign tries the position in discrete steps (probably based on baseline distance) and does not strive to achieve ideal geometric fitting.

Although the ability of desktop publishing tools to format text in general shapes is not new, Adobe InDesign's paragraph composer brings an extra quality. Unlike its predecessors it tries to optimize visual appearance of the paragraph as a whole. The criteria are very similar to those used by T_EX . Taking into account also its facility of optical kerning and margin alignment, the traditional argument about T_EX dominance in typographic quality is well challenged. It is advisable to consider, that, for certain tasks, the application of Adobe InDesign might be more productive with comparable or better results than using T_EX .

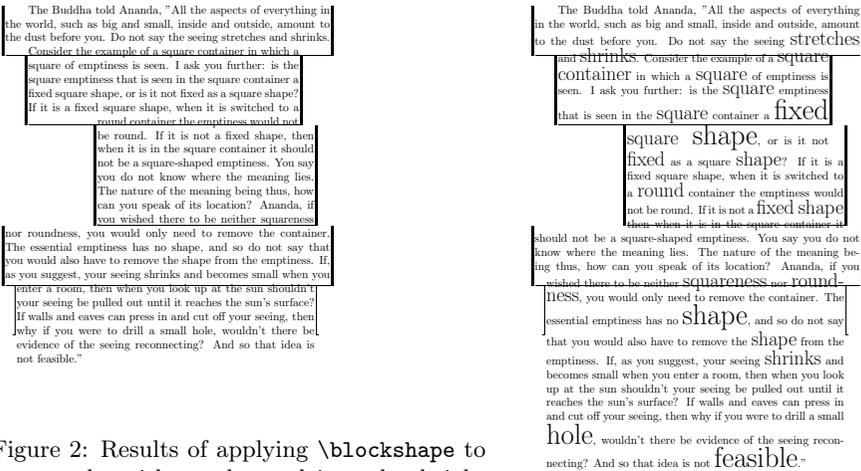


Figure 2: Results of applying `\blockshape` to paragraphs with regular and irregular height of lines. The auxiliary lines indicating the blocks were added by T_EX macro programming.

New `\blockshape` primitive

Although the author's more ambitious aim is to unify text and graphics models and to provide for a general specification of text shapes in terms of graphic paths (see [11]), for this experiment he has chosen a simple and pragmatic solution. The new `\blockshape` primitive is quite similar to `\parshape` and behaves exactly in the same way with only one exception: the indents and widths are not specified for the individual lines but rather for subsequent blocks of certain height. The syntax of the command is:

```
\blockshape N
    indent1 width1 height1
    indent2 width2 height2
    ...
    indentN widthN
```

Note that the last line in the list omits the height parameter because the last indent and width run for the rest of the paragraph (analogically to the `\parshape`). The application of the new primitive is illustrated by Figure 2.²

In order to understand how this new feature is implemented it is necessary to take a closer look at the line-breaking algorithm and data structures.

Abstract model of the line-breaking

The $\text{T}\mathcal{E}\text{X}$ algorithm for breaking paragraphs into lines is described in [8] and [7]. We will focus on its central part and try to explain it on an abstract level. The input of the algorithm is a paragraph given as one long line of text consisting of characters, boxes, and other special items. The output is an increasing sequence of positions. The resulting lines are obtained by breaking the input line at these positions. The algorithm uses the following graph model:

- Set of nodes N .

Node $m \in N$ represents a valid breakpoint in the paragraph, it means the place where it is legal to end one line and to start a new one. The set N covers all such potential breakpoints. The nodes are ordered. Let $break(n)$ be the breakpoint represented by node n , then for $m, n \in N$ we put $m < n$ iff $break(m)$ is closer to the beginning of the paragraph than $break(n)$ (and the relation \leq is $< \cup id$ as usual). The set contains two special nodes $a, z \in N$ such that $break(a)$ lies at the beginning and $break(z)$ at the end of the paragraph.

- Set of directed edges $E \subseteq N \times N$.

Edge $(m, n) \in E$ starting at node m and ending at node n represents the line of text delimited by breakpoints $break(m)$ and $break(n)$. A line has a certain natural width and thanks to elastic spaces it can shrink or stretch to a certain extent. For each $(m, n) \in E$ we have $m < n$, so the graph is acyclic.

²In the left example with uniform line height, it looks as if some lines should be longer than they are. That is because the current implementation works simply with the whole bounding boxes of the individual lines. Therefore a line belongs to the narrower block even if only the top or the bottom of the bounding box is intersected. Taking graphical envelopes of the lines into account could give more appropriate results but that is far beyond the current $\text{T}\mathcal{E}\text{X}$ and $\mathcal{N}\mathcal{T}\mathcal{S}$ possibilities.

- Set of paths $P \subseteq \bigcup_{i=1}^{|N|} N^i$.
 Path $p \in P$ is a sequence of nodes $[n_0, n_1, \dots, n_k]$ such that $n_0 = a$ and $[n_i, n_{i+1}] \in E$ for $i = 0, \dots, k - 1$. The length of p , denoted $|p|$, is k . All the paths in our model start at a , because we do not need to consider anything else.
- Function *desired*: $P \rightarrow Real$ assigns a certain dimension value to a path. The meaning is the desired width of the last line in the sequence of lines represented by that path. In current implementation, the value depends only on the settings of `\parshape`, `\hangindent` or `\hsize` and the ordinal number of the last line ($|p|$).
- Function *price*: $E \rightarrow Real \rightarrow Num$ assigns a certain numeric value to a combination of line and its desired width. The value depends on how much the line has to be stretched or shrunk to fit into the width and how “expensive” its ending breakpoint was.
- Function *extra*: $P \rightarrow Num$ assigns a numeric value to a path. It represents “extra” costs associated with the last edge on the path which can be induced by non-matching lines (a tight line is followed by a loose one or vice versa), two hyphens in a row, or preferences for a certain total number of lines (arises at the end if `\looseness` is active).
- Function *cost*: $P \rightarrow Num$ represents the total cost of the path. Let $p = [n_0, \dots, n_k]$ be a path and $p[j] = [n_0, \dots, n_j]$ for $j \leq k$ its subpath of length j . Then the *cost* of p is defined as:

$$cost(p) = \sum_{i=1}^k [price((n_{i-1}, n_i), desired(p[i])) + extra(p[i])].$$

We see that the model is quite special because the cost of a path depends on many factors and not only on fixed prices of individual edges as usually.

The goal of the algorithm is to find the optimal path $o = [a, \dots, z]$ with minimal $cost(o)$. There are potentially $2^{|N - \{a, z\}|}$ paths leading from beginning to end so the complexity of the algorithm is potentially very high. The current implementation uses some smart optimizations and its complexity is proportional only to $|N|W$, where W is the maximal number of potential breakpoints per line.

The algorithm used by $T_{\text{E}}X$ works by visiting all nodes $m \in N$ in ascending order and by keeping track of all relevant paths $[a, \dots, n]$ such that $n \leq m$. The first and main optimization is based on early discarding of paths with

infinite costs. The algorithm simply does not explore the edges representing lines which are too long to fit in a desired width and their successors.

The basic idea of the second optimization is that, if several paths lead to the same node then only the one with the minimal cost is relevant for future prolongation. The basic complication of this method is that the paths can have different influences on the future costs along the way. For that reason the model introduces a relation of equivalence $equiv \in P \times P$. If it is known in advance, that the future costs induced by two paths will be equal, then the paths are considered to be equivalent in the relation $equiv$. This happens when the corresponding last lines have the same “fitness” (tight, decent, loose, ...) and the following sequences of desired line widths will be the same (depending on `\parshape`) as well as the final extra cost (imposed by `\looseness`). Having the equivalence $equiv$, only the best paths from each factorized subset are kept. Further on, TEX knows the maximal difference of future costs for each pair of subsets and discards the whole subset if it has no chance to compete in future comparisons.

Given this abstract graph model, what do we have to change in order to extend the algorithm to handle general graphic shapes of the paragraph? The main difference is in the function *desired*. While in the current shaping method based on `\parshape`, the value of *desired* depends only on a number of previous lines, in a general graphic case it depends on the actual vertical position and height of the line in the shape. This position surely depends on heights of the previous lines. The relation $equiv$ is influenced in a similar way. If we restrict ourselves to a simple case (employed in InDesign) with no vertical elasticity of lines and a limited number of possible positions of the first line, then it should be enough just to modify *desired* and $equiv$. Another modification would be necessary for handling holes inside shapes but it is mainly a matter of arranging lines after breaking.

If we admit vertical shrinkability and/or stretchability in paragraphs to be broken, then the whole situation changes significantly. The desired widths of the lines may change when they are finally adjusted vertically and this depends on the whole content of the shape and not just on the current paragraph. The function *desired* (and $equiv$) then depend not only on the path passed so far but also on complex future factors. The currently used optimizations would not work any more and we would have to try to invent new ones.

Also the positioning of the first line in an irregular shape is quite delicate. It can fit into different vertical offsets if it is differently stretched or shrunk and finding the minimal cost with respect to the following lines is probably next to impossible.

The implementation of `\blockshape`

After the abstract model was understood, the implementation was quite straightforward. The core of the line-breaking algorithm in $\mathcal{N}\mathcal{T}\mathcal{S}$ was implemented in the class `Breaker`³. In order to allow for alternative algorithms and/or implementations, only the extracted abstract interface of the class was left under the name `Breaker` and the original class was renamed to `BaseBreaker` and declared to implement `Breaker` interface. The creation of the instance of the concrete class `BaseBreaker` was moved from the code of `Paragraph` to a factory method of the class `TypoCommand.Config` so that the alternative instance can be plugged-in on the very top configuration level without modifying the productive code.

The rest of the new functionality was implemented outside of the $\mathcal{N}\mathcal{T}\mathcal{S}$ code in the extending package called $\mathbf{X}\mathcal{T}\mathcal{S}$. The first step consisted in providing an alternative open implementation for the original algorithm. It was separated to three levels. The basic class `BaseOpenBreaker` reflects the abstract model described above and leaves the implementation of the functions *desired*, *extra*, and relation *equiv* to its refining subclasses. The class `FitnessOpenBreaker` on the next level defines the function *extra* and relation *equiv* according to the standard rules for line fitness, etc. The class `LinesShapeBreaker` on the most concrete level defines the function *desired* based on the standard `\parshape` and `\hang*` calculations.

The code implementing the new functionality itself was surprisingly small. The class `BlockShape` keeps the data of the block-shape description and defines a method which provides the indent and width for a box of certain position and height. An instance of this class is created by a new command `BlockShapeParam` which is a simple analogy to `ParShapeParam` from $\mathcal{N}\mathcal{T}\mathcal{S}$. The descriptor is used by the class `BlockShapeBreaker` which is a subclass of `FitnessOpenBreaker` (sibling of `LinesShapeBreaker` in the inheritance hierarchy). The basic idea of `BlockShapeBreaker` is that it refines the nodes

³For the sake of readability the locations of the classes inside of the package hierarchy (full class names) are omitted. The names are unambiguous and can be easily located in the source tree.

so that they keep information about the position and dimensions of the preceding line and the desired width is then based on fitting into block-shape instead of just the line number.

The main application level of $\mathcal{X}\mathcal{T}\mathcal{S}$ provides its own configuration which defines also the extension features. If a block-shape is defined for a paragraph in question then it instantiates `BlockShapeBreaker`, otherwise it falls back to `LinesShapeBreaker`.

Evaluation of the new feature

The `\blockshape` primitive, as it is defined now, is only a very first and experimental step toward the formatting of a text into a general graphical shape. Although the author proposes the introduction of graphics structures into the current box-based $\mathcal{T}\mathcal{E}\mathcal{X}$ document model (see [11]), the specification by an orthogonal polygon can effectively approximate many graphical shapes. The restriction of the block-shape specification is that it neither provides for holes nor for non-convex top and bottom. This limitation could still be solved using orthogonal polygons. In the discussion of the abstract model it was already mentioned that it would affect rather the arrangement of the lines after breaking than the core algorithm itself.

The most serious restriction which limits the practical use is that no vertical glue is allowed inside the paragraph. This problem is non-trivial and requires a much more sophisticated solution. Note that the same restriction applies in Adobe InDesign.

Another impractical aspect of the experimental feature is its current paragraph-only scope (analogically to `\parshape`). The block-shape of the text actually starts on a position in the text flow rather than on a fixed place on a page. Even deciding the position of the first line in the block-shape was not clear at the beginning. From several possibilities based on relations between baselines and `\prevdepth`, the author eventually chose probably the simplest one: the top-edge of the block-shape matches the top-edge of the first line. This works fine at the beginning of a page or of a vertical box but it is perhaps not so useful in the middle. Its practical application for page-layout would need a more natural blending with the usual document processing. Any suggestion is welcome.

The main value of the experiment consists in showing the ease of changing and extending $\mathcal{N}\mathcal{T}\mathcal{S}$. Devising the abstract model of line-breaking described above

took about one week during the writing of [11]. The whole implementation took only two hard day-nights at the EuroBachTeX'2002 conference.

Extending versus changing

The ideal situation for making extensions and changes to the behavior of $\mathcal{N}\mathcal{T}\mathcal{S}$ would be if there was no need to touch the existing code. Although $\mathcal{N}\mathcal{T}\mathcal{S}$ is pretty modular (one of the testers declared that it is over-modularized), this ideal is unfortunately not reached. During the reimplementing process, the main focus was on the faithful simulation of the original TeX and the ways how to achieve the best possible extendibility were only gradually learned. The greatest contribution and basic precondition to that goal was the layered architecture where the lower layers of $\mathcal{N}\mathcal{T}\mathcal{S}$ do not depend on the upper ones.

The main factor which forces modifications of the existing code is the instantiation of concrete classes. It is beneficial to employ the creational design patterns (see [2]) to avoid such kind of dependencies. In $\mathcal{N}\mathcal{T}\mathcal{S}$, these and similar design patterns are used quite often⁴, mainly for avoiding cyclic dependencies and enabling the hierarchy of layers. But for the ideal extendibility without the need of modifications of the original code, it would be necessary to use *abstract factory* or *factory method* for nearly all object creations because nobody knows what is going to be extended in the future. And there are other problems as well, hence it is not clear at all whether the ideal extendibility could be achieved even after a major revision and systematic application of the patterns.

In this context, it seems that the decision of the DANTE e.V. board to publish $\mathcal{N}\mathcal{T}\mathcal{S}$ under a license analogical to the L^AT_EX Project Public License which disallows modifications of the original code was perhaps not really fortunate. In the L^AT_EX project, the requirement of the license is not a problem because, in the system implemented as a collection of TeX input files and with the provision for file aliases, it is quite easy to redefine a macro or to replace a file containing TeX commands. Java code has a much more complicated structure and the file names must correspond to the classes they define, therefore such simple textual redefinitions are not possible. However, the $\mathcal{N}\mathcal{T}\mathcal{S}$ Public License assumes keeping the original untouched and that is not realistic. After all, it is not clear whether the author himself is formally allowed to modify $\mathcal{N}\mathcal{T}\mathcal{S}$.

⁴The author became aware of the names of the patterns and was hinted to [2] only after they were applied during the major period of the reimplementing.

For both legal and technical reasons, the author decided to keep $\mathcal{N}\mathcal{T}\mathcal{S}$ relatively closed and stable and to conduct all new experiments in a separate package called $\mathfrak{X}\mathfrak{T}\mathfrak{S}$ (\mathfrak{X} for eXtension, eXperimental, eX- $\mathcal{N}\mathcal{T}\mathcal{S}$, ...). It will still be necessary to make changes to $\mathcal{N}\mathcal{T}\mathcal{S}$ but the modifications can merely open a door to a separate extension so they usually need not be very large. In this way, $\mathcal{N}\mathcal{T}\mathcal{S}$ will always provide the same canonical functionality.

Conclusion

The Java-based implementation of $\mathcal{N}\mathcal{T}\mathcal{S}$ is maybe too slow to become the basis of the real typesetting system of the future. But, if you want to try out your bright ideas about features which the future system should have, then $\mathcal{N}\mathcal{T}\mathcal{S}$ (with $\mathfrak{X}\mathfrak{T}\mathfrak{S}$) is the alternative to the Pascal WEB implementation of $\mathbb{T}\mathbb{E}\mathbb{X}$. In order to make a fair comparison of the ease of implementing a new feature in both systems, it would be necessary to try introducing `\blockshape` into the original $\mathbb{T}\mathbb{E}\mathbb{X}$ by somebody who is really familiar with the $\mathbb{T}\mathbb{E}\mathbb{X}$ code. Also the possibility to combine several extensions and to provide features optionally should be taken into account. Does anybody want to try?

For the author of the code, it must be certainly much easier to make changes and extensions, especially when the documentation is mostly missing. For those who want to try, the author is prepared to discuss and give advice for the implementation of their ideas.

It may be useful to turn $\mathfrak{X}\mathfrak{T}\mathfrak{S}$ into a properly maintained open-source project. But this is too much work for just one pair of hands. Is anybody willing to help?

In the end, let the author express his thanks to all people who provided him a motivation for his work on $\mathcal{N}\mathcal{T}\mathcal{S}$ and $\mathfrak{X}\mathfrak{T}\mathfrak{S}$ and to those who will do so in the future.

References

- [1] Adobe Systems Incorporated; *Adobe InDesign 2.0 User Guide for Windows and Macintosh*; 2001.
- [2] Erich Gamma, Richard Helm et al.: *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison Wesley Publishing Company; 1995.

- [3] Michel Goossens, Frank Mittelbach und Alexander Samarin: *The L^AT_EX Companion*; Addison Wesley Publishing Company; Reading, Massachusetts; 1993.
- [4] Hans Hagen: *Some N_TS thoughts*; in *Proceedings of EuroT_EX'99*; September 1999.
- [5] Hans Hagen: *ConT_EXt the Manual*; November 2001; <http://www.pragma-ade.com/document.htm>.
- [6] Donald E. Knuth: *The T_EX Book*; Addison Wesley Publishing Company; Reading, Massachusetts; 1986.
- [7] Donald E. Knuth: *T_EX The Program*; Addison Wesley Publishing Company; Reading, Massachusetts; 1986.
- [8] Donald E. Knuth und Michael F. Plass: *Breaking Paragraphs Into Lines; Software—Practice and Experience*; 11(11); November 1981.
- [9] Frank Mittelbach: *E-T_EX: Guidelines for Future T_EX Extensions; TUGBoat*; 11(3), S. 337–345; September 1990.
- [10] Simon Pepping: *Extending ExT_EX*; in *EuroT_EX 2001 Proceedings*; S. 146–150; September 2001.
- [11] Karel Skoupý: *Proposal for Text and Graphics Unification in a T_EX-Based Framework*; in *Proceedings of EuroBachoT_EX'2002*; S. 81–87; May 2002.
- [12] Karel Skoupý und Taylor Philip: *The Implementation of N_TS*; in *EuroT_EX'99 Proceedings*; S. 246–260; September 1999.

Tipps und Tricks: Tierkunde einmal anders

Rolf Niepraschk

Die allseits bekannte L^AT_EX-Umgebung `description` wird zum Setzen von hervorgehobenen Begriffen und zugehörigen Beschreibungstexten verwendet. Das Ergebnis sieht, je nach verwendeter Dokumentenklasse, ähnlich dem folgenden vorwiegend aus [4] entnommenen Beispiel aus:

Gämse: ein großes Tier, das westlich des Semmering von Touristen verjagt wird.

Gelse: ein kleines Tier, das östlich des Semmering Touristen verjagt.

Gürteltier: ein mittelgroßes Tier, das hier nur wegen der Länge seines Namens vorkommt.

Südostasiatisches Hängebauchschwein: ein sympathisches Tier mit einem noch längeren Namen.

Es könnte der Wunsch aufkommen, dass sich die Einrückung der Beschreibungstexte nach dem längsten Begriff richtet, sofern dieser keine zu große Breite aufweist. Es existieren zu diesem Zweck Pakete, wie z. B. [1, 3], die modifizierte `description`-Umgebungen definieren. Bei ihnen kann mit zusätzlichen Parametern eine einheitliche Einrückung erzielt werden. Nachteilig ist, dass es nicht automatisch geschieht. Das Paket `eqlist` [5] bietet eine bessere Lösung. Die im Folgenden gezeigte umdefinierte `description`-Umgebung rückt automatisch in gewünschter Weise ein:

```
\usepackage{eqlist}[2002/08/15]

\renewenvironment{description}
{\begin{eqlist}
[%
  \eqlistauto{.33\linewidth}
  \def\makelabel##1{\hspace{-\labelsep}\descriptionlabel{##1}}
]}
{\end{eqlist}}
```

Das am Anfang gezeigte Beispiel sieht mit der neuen Definition nun so aus:

Gämse: ein großes Tier, das westlich des Semmering von Touristen verjagt wird.

Gelse: ein kleines Tier, das östlich des Semmering Touristen verjagt.

Gürteltier: ein mittelgroßes Tier, das hier nur wegen der Länge seines Namens vorkommt.

Südostasiatisches Hängebauchschwein: ein sympathisches Tier mit einem noch längeren Namen.

Intern lädt `eqlist` das Paket `eqparbox` [2], um den Text der Begriffe in jeweils gleichbreite Boxen zu packen. Das folgende Beispiel zeigt vereinfacht das in `eqlist` verwendete Prinzip:

```
\usepackage{eqparbox}
```

```
\fbox{\eqparbox{HUGO}{G"amse:}}\fbox{\eqparbox{HUGO}{Gelse:}}
\fbox{\eqparbox{HUGO}{G"urteiltier:}}
```

Das Ergebnis ist:

Gämse:	Gelse:	Gürteltier:
--------	--------	-------------

Alle Boxen, die der Gruppe „HUGO“ angehören, erhalten automatisch die Breite des breitesten Gruppenmitglieds. Die `eqlist`-Umgebung verwendet intern ebenfalls eine gemeinsame Gruppe für alle Parameter der `\item`-Anweisungen, wobei `\eqlistauto{.33\linewidth}` dafür sorgt, dass das „Hängebauchschwein“ wegen Überbreite ($> 0,33 \times \text{linewidth}$) von der gemeinsamen Gruppe ausgeschlossen wird. Seine Breite bleibt für die Einrückung unberücksichtigt.

Für dieses interessante Konzept findet man sicher auch in anderem Zusammenhang lohnende Anwendungen.

Literatur

- [1] Rainer Hülse und Wolfgang Kaspar: *EXPDLIST*; Sept. 1999; CTAN: `tex-archive/macros/latex/contrib/supported/expdlist/expdlist.dtx`.
- [2] Scott Pakin: *The eqparbox package*; Apr. 2001; CTAN: `tex-archive/macros/latex/contrib/supported/eqlist/eqparbox.dtx`.
- [3] Alexander Rozhenko: *Desclist Package*; Jan. 2002; CTAN: `tex-archive/macros/latex/contrib/supported/ncctools/desclist.sty`.
- [4] Walter Schmidt, Jörg Knappen et al.: *L^AT_EX 2_ε-Kurzbeschreibung*; Juni 2001; CTAN: `tex-archive/info/lshort/german/l2kurz.pdf`.
- [5] Martin Väth: *The eqlist package*; Aug. 2002; CTAN: `tex-archive/macros/latex/contrib/supported/eqlist/eqlist.dtx`.

gauss.sty – Visualisierung des Gaußschen Eliminationsverfahrens

Herbert Voß

Die Zahl der vorhanden \LaTeX -Pakete ist sicherlich endlich, wenn man auch manchmal den gegenteiligen Eindruck hat. Neben den mehr oder weniger bekannten Paketen gibt es eine erhebliche Anzahl, die einem großen Anwenderkreis deswegen nicht bekannt sind, weil deren Anwendung häufig sehr speziell ist. Dazu gehört sicherlich das Paket `gauss`, welches nur sehr spezielle Wünsche erfüllt, die im Folgenden an Beispielen beschrieben werden.

Einführung

Das Gaußsche Eliminationsverfahren zur Lösung linearer Gleichungssysteme kann als Standardanwendung innerhalb der Naturwissenschaften bezeichnet werden. \TeX ist dafür bekannt, dass es insbesondere den mathematischen Zeichensatz hervorragend beherrscht. Dabei kann es in der Darstellung trotzdem zu erheblichen Problemen kommen, wenn zusätzliche Erläuterungen zu den einzelnen Rechenschritten erforderlich sind.

Für diesen Fall bietet das Paket `gauss` eine spezielle Unterstützung. Das Paket kann von jedem beliebigen CTAN-Server heruntergeladen werden (`CTAN:macros/latex/contrib/other/gauss/gauss.sty`), beispielsweise von `ftp.dante.de`. Eine entsprechende, ausführliche Dokumentation mit einem zusätzlichen Beispiel im Quellcode findet sich ebenfalls dort. Das Paket wird in der üblichen Art und Weise in einen bestehenden \TeX -Baum eingebunden. Das Paket `gauss` verwendet zum Erstellen der diversen Linienzüge ausschließlich die Umgebung `picture`, sodass eine PDF-Ausgabe ohne weiteres möglich ist.

gmatrix-Umgebung

In Analogie zum Paket `amsmath` wird eine Matrixumgebung zur Verfügung gestellt, was an sich nichts besonderes ist, wenn man einmal davon absieht, dass standardmäßig keine Klammerung vorgesehen ist. Diese kann der Einfachheit halber über einen optionalen Parameter gesetzt werden, der letztlich bestimmt, welche der Matrixumgebungen von `amsmath` benutzt wird:

Option	Bedeutung
	$\backslash\begin{gmatrix} \rightarrow \backslash\begin{matrix} \rightarrow \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \end{matrix}$
v	$\backslash\begin{gmatrix}[v] \rightarrow \backslash\begin{vmatrix} \rightarrow \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$
V	$\backslash\begin{gmatrix}[V] \rightarrow \backslash\begin{Vmatrix} \rightarrow \begin{Vmatrix} 1 & 2 \\ 3 & 4 \end{Vmatrix}$
p	$\backslash\begin{gmatrix}[p] \rightarrow \backslash\begin{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$
b	$\backslash\begin{gmatrix}[b] \rightarrow \backslash\begin{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
B	$\backslash\begin{gmatrix}[B] \rightarrow \backslash\begin{Bmatrix} \rightarrow \begin{Bmatrix} 1 & 2 \\ 3 & 4 \end{Bmatrix}$

Ohne Parameter ist es die einfache Matrixumgebung von `amsmath`. Daneben wird noch der Befehl `\newmatrix` angeboten, der es ermöglicht, weitere Umgebungen mit beliebigen Klammern zu definieren, beispielsweise:

$$\backslash\newmatrix{[]}\{[]\}{q} \rightarrow \backslash\begin{gmatrix}[q] \rightarrow \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$$

Die ersten beiden Parameter geben die Klammerung und der dritte das Options-Kürzel an. Es ist offensichtlich, dass als Kürzel `v`, `V`, `p`, `b`, `B` wenig Sinn machen und `g` nicht zur Verfügung steht.

Zeilen- und Spaltenmarkierungen

Die Zählung der Zeilen und Spalten beginnt in der linken oberen Ecke jeweils bei 0, womit der rechten unteren Ecke dann (Zeilenanzahl-1, Spaltenanzahl-1) zugeordnet wird.

Zeilenmarkierungen werden durch den Befehl `\rowops` und Spaltenmarkierungen durch `\colops` eingeleitet. Für die speziellen Markierungen (Operationen) stehen die folgenden drei Befehle zur Verfügung:

$$\begin{bmatrix} 1 & -2 & 1 & 2 \\ 2 & 3 & -2 & 3 \\ 4 & -1 & 3 & -1 \\ 3 & 2 & -4 & 5 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = \begin{bmatrix} 8 \\ 14 \\ 7 \\ 5 \end{bmatrix} \quad (1)$$

$$\begin{array}{cccc|cccc} 1 & -2 & 1 & 2 & 8 & | \cdot (-2) & | \cdot (-4) & | \cdot (-3) \\ 2 & 3 & -2 & 3 & 14 & \longleftarrow & + & \\ 4 & -1 & 3 & -1 & 7 & \longleftarrow & + & \\ 3 & 2 & -4 & 5 & 5 & \longleftarrow & + & \end{array}$$

$$\begin{array}{cccc|cccc} 1 & -2 & 1 & 2 & 8 & & & \\ 0 & 7 & -4 & -1 & -2 & | \cdot (-1) & | \cdot (-\frac{8}{7}) & \\ 0 & 7 & -1 & -9 & -25 & \longleftarrow & + & \\ 0 & 8 & -7 & -1 & -9 & \longleftarrow & + & \end{array}$$

$$\begin{array}{cccc|cccc} 1 & -2 & 1 & 2 & 8 & & & \\ 0 & 7 & -4 & -1 & -2 & & & \\ 0 & 0 & 3 & -8 & -23 & | \cdot \frac{17}{3} & & \\ 0 & 0 & -17 & 1 & -47 & \longleftarrow & + & \end{array}$$

$$\begin{array}{cccc|cccc} 1 & -2 & 1 & 2 & 8 & & & \\ 0 & 7 & -4 & -1 & -2 & & & \\ 0 & 0 & 3 & -8 & -23 & & & \\ 0 & 0 & 0 & -\frac{133}{3} & -\frac{532}{3} & & & \end{array}$$

Die vertikale Ausrichtung ist dabei nicht ganz unproblematisch, denn einerseits muss alles linksbündig ausgerichtet sein und andererseits haben die einzelnen Zellen eine unterschiedliche horizontale Breite. In solchen Fällen erstellt man am besten nicht vier verschiedene Matrizen, sondern setzt alles in eine einzige „gmatrix“-Umgebung und markiert sich dabei einfach die einzelnen Zeilen:

```
\begin{gmatrix}
  1 & -2 & 1 & 2 & & 8 & \\ % Zeile 0
  2 & 3 & -2 & 3 & & 14 & \\ % Zeile 1
  4 & -1 & 3 & -1 & & 7 & \\ % Zeile 2
  3 & 2 & -4 & 5 & & 5 & \\ % Zeile 3
  & & & & & & \\ % Zeile 4
  1 & -2 & 1 & 2 & & 8 & \\ % Zeile 5
\end{gmatrix}
```



```

\begin{tabular}{p{\columnwidth}}
\begin{equation}
\begin{gmatrix}[p]
1 & -2 & 1 & 2 & \backslash \ % \text{Zeile 0}
2 & 3 & -2 & 3 & \backslash \ % 1
4 & -1 & 3 & -1 & \backslash \ % 2
3 & 2 & -4 & 5 & \ % 3
\rowops
\mult{0}{(-2)}\add{0}{1}
\mult{0}{(-4)}\add{0}{2}
\mult{0}{(-3)}\add{0}{3}
\end{gmatrix}
\end{equation}
\\ % Ende 1. Matrix
\begin{equation}
\begin{gmatrix}[p]
1 & -2 & 1 & 2 & \backslash \ % \text{Zeile 0}
0 & 7 & -4 & -1 & \backslash \ % 1
0 & 7 & -1 & -9 & \backslash \ % 2
0 & 8 & -7 & -1 & \ % 3
\rowops
\mult{1}{(-1)}\add{1}{2}
\mult{1}{(-\frac{8}{7})}\add{1}{3}
\end{gmatrix}
\end{equation}
\end{tabular}

```

Insbesondere die Funktion `mult` kann auch für normale Gleichungsumformungen benutzt werden, wenn man diese innerhalb der `gmatrix`-Umgebung beschreibt.

$$\begin{aligned}
 \frac{1}{C} \int i dt + L \cdot \frac{di}{dt} + R \cdot i &= 0 \quad | \text{Differentiation} \\
 \frac{1}{C} \cdot i + L \cdot \frac{d^2i}{dt^2} + R \cdot \frac{di}{dt} &= 0 \quad | : L \text{ und sortieren} \\
 \frac{d^2i}{dt^2} + \frac{R}{L} \cdot \frac{di}{dt} + \frac{1}{LC} \cdot i &= 0 \quad \left| \frac{R}{L} = \delta; \frac{1}{LC} = \omega_0^2 \right. \\
 \ddot{i} + \delta \dot{i} + \omega_0^2 i &= 0
 \end{aligned} \tag{4}$$

Gleichung 4 ergibt sich durch folgende Befehlssequenzen, wobei der erste Befehl im folgenden Abschnitt beschrieben wird:

```
\renewcommand\rowmultlabel[1]{\left|,\,#1\right.}
\begin{equation}
  \displaystyle
  \frac{1}{C}\int i\cdot dt+L\cdot\frac{di}{dt}+R\cdot i\hfill & = & 0\backslash
  [ \dots ]
  \displaystyle
  \ddot{i}+\delta\dot{i}+\omega_0^2i\hfill & = & \&0
\rowops
\mult{0}{\mbox{Differentiation}}
\mult{1}{:L\mbox{ und sortieren}}
\mult{2}{\displaystyle\frac{R}{L}=\delta;\ \frac{1}{LC}=\omega_0^2}
\end{gmatrix}
\end{equation}
```

Weitere Funktionen

Das Paket `gauss` gestattet die Neudefinition der Label für die Additions- bzw. Multiplikationsmarkierungen, sodass beispielsweise ein Multiplikationspunkt standardmäßig gesetzt werden kann.

```
\renewcommand{\rowmultlabel}[1]{|\,\cdot\#1}
```

Für die Zeilen und Spalten gibt es dabei unterschiedliche Label, die in der üblichen Weise überschrieben werden können.

```
\rowaddlabel
\rowmultlabel
\coladdlabel
\colmultlabel
```

Hinweise

Bei der Anwendung der Umgebung `gmatrix` ist im Wesentlichen nur zu beachten, dass die letzte Zeile keinen `\-`Befehl aufweisen darf. Die letzte Zeile wird auch für die Berechnung der einzelnen Spaltenbreiten herangezogen, woraus folgt, dass alle Spalten vorhanden sein müssen. Die jeweilige Höhe einer Zeile orientiert sich nur an der Höhe der einzelnen Spalten, nicht jedoch an der Höhe der Label.

Es ist grundsätzlich darauf zu achten, dass bei gleichzeitiger Verwendung von `amsmath` dieses vor `gauss` geladen wird.

Hüllen für Musikkassetten

Florian Benischke

Mit Hilfe von `mceinleger.sty` lassen sich Hüllen und Einleger für Musikkassetten erstellen. Dieser Artikel soll dieses Paket vorstellen und seine Benutzung erleichtern. Unter „Hülle“ wird in diesem Zusammenhang eine Einlage für Musikkassetten verstanden, die aus einer Vorder-, Schmal- und Rückseite besteht, ein „Einleger“ umfasst zusätzlich noch eine einklappbare Seite, die an die Vorderseite anschließt.

Einleitung

Um anderen L^AT_EX-Nutzern die „Bastelarbeit“ zu ersparen, wurde mit dem Paket `mceinleger.sty` (CTAN: `tex-archive/macros/latex/contrib/supported/mceinleger/mceinleger.sty`) die Grundlage geschaffen, selbst Hüllen und Einleger für Musikkassetten zu produzieren. Bei der Gestaltung einer Hülle lässt sich die gewünschte Information auf drei Seiten aufteilen:

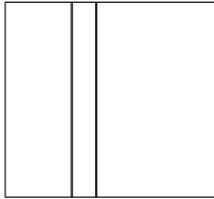
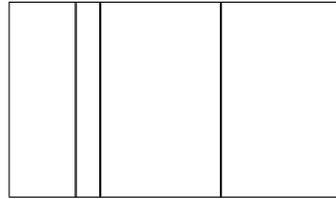
- Auf der Vorderseite empfiehlt es sich, mit Formatierungen wie beispielsweise `\huge` den Titel entsprechend hervorzuheben, ebenso kann mit der Umgebung `description` eine genaue Inhaltsangabe der beiden Kassettenseiten (A und B) erfolgen.
- Die Schmalseite sollte als im Regal sichtbarer Teil den prägnanten Titel enthalten.
- Die Rückseite kann je nach Bedarf den Inhalt der Kassette in geraffter Form noch einmal wiedergeben.

Wichtig: Dieses Paket baut auf `rotating.sty` auf, es muss daher zusätzlich noch geladen werden, damit `mceinleger.sty` funktionieren kann.

Benutzung des Pakets

Zwei Möglichkeiten stehen zur Auswahl: Der Befehl `\huelle` erzeugt die Anordnung, die in Abbildung 1(a) zu sehen ist.

Der Befehl `\einleger` erlaubt zusätzlich noch die Gestaltung einer weiteren Seite, die durch Einklappen einen MC-Einleger erzeugt. Dies ist in Abbildung 1(b) zu sehen.

(a) `\huelle`(b) `\einleger`

Die zusätzliche Seite des Einlegers gestattet unter anderem eine genaue Auflistung des Inhalts (mit genauer Dauer etc.). Die Syntax dieser beiden Befehle lautet dabei wie folgt:

```
\huelle{A}{B}{C}{D}
```

Dabei gilt:

A Text der Rückseite

B Text der seitlich sichtbaren Seite (= Schmalseite)

C Text der Vorderseite

D Anordnung des Textes auf der Vorderseite; folgende Möglichkeiten stehen zur Verfügung: top (t), bottom (b), centered (c), stretched (s), left (l) und right (r).

Dabei sind auch Kombinationen wie beispielsweise `tl` für top-left zulässig.

```
\einleger{A}{B}{C}{D}{E}{F}
```

Dabei gilt:

A Text der Rückseite

B Text der seitlich sichtbaren Seite (= Schmalseite)

C Text der Vorderseite

D Anordnung des Textes auf der Vorderseite

E Text auf der einklappbaren Seite

F Anordnung des Textes auf der einklappbaren Seite

Zusammenfassung

Unter Mitverwendung des Pakets `rotating.sty` ermöglicht `mceinleger.sty` die Herstellung von selbstgestalteten Hüllen und Einlegern von Musikkassetten. Hüllen bestehen dabei aus Vorder-, Schmal- und Rückseite, mit dem

Befehl `\einleger` kann auch noch eine an die Vorderseite angefügte, einklappbare Seite mit detailliertem Inhaltsverzeichnis oder Ähnlichem erstellt werden.

Tipps und Tricks: Fette Schreibmaschinenschrift

Rolf Niepraschk

Gelegentlich besteht der Wunsch, eine fette Schreibmaschinenschrift zum Hervorheben einzelner Worte einzusetzen. So ist es nützlich, dies bei der Darstellung von Programmtexten zu tun, um beispielsweise Schlüsselworte der betreffenden Programmiersprache zu kennzeichnen. Das \LaTeX -Paket `listings` bietet die Möglichkeit, solche Schlüsselworte und andere Besonderheiten einer Vielzahl von Programmiersprachen entsprechend aufzubereiten. Die Auswahl an Schreibmaschinenschriften ist jedoch nur gering; die fetten Varianten sind besonders rar. Standardmäßig bieten die meisten \TeX -Distributionen keine fette Variante der zu den Computer-Modern-Schriften gehörenden Schreibmaschinenschrift an. Zwar existiert eine solche (`cmbtt`), und in der Anleitung zum `listings`-Paket [1] wird auch beschrieben, wie sie nutzbar gemacht werden kann, trotzdem ist das eher eine Notlösung. Die nichtfette Variante ist bereits recht kräftig, und notgedrungen wirkt die fette zu sehr auffallend. Noch stärker ins Gewicht fällt aber der Umstand, dass sie nicht im Type1-Format existiert. Beim Erzeugen von PDF-Dateien ist das ein ernsthafter Mangel. Eine andere sowohl in fetter Variante als auch im Type1-Format verfügbare Schrift ist `Courier`. Allerdings ist auch sie keine ideale Lösung, da sie nicht gut zu den anderen üblicherweise verwendeten Schriftschnitten passt.

Mit Einführung der Version 4.2 von `XFree86`¹ wurde der Allgemeinheit eine neue frei verwendbare Schreibmaschinenschrift im Type1-Format zur Verfügung gestellt, womit es auch eine gute Alternative für den Textsatz gibt. Ihr Name ist `LuxiMono`. Um sie mit \LaTeX verwenden zu können, ist am Anfang des Dokuments das Paket `luximono` [2] zu laden. Das folgende Beispiel zeigt die Verwendung der Schrift `LuxiMono` zusammen mit dem `listings`-Paket:

```
\documentclass{article}
\usepackage[T1]{fontenc}
```

¹Die freie Implementierung der unter Linux üblichen Grafikoberfläche

```

\usepackage[scaled]{luximono}
\usepackage{listings}
\begin{document}
\lstset{language=C,basicstyle=\ttfamily,
commentstyle=\itshape,frame=single}
\begin{lstlisting}
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    /* Ein simples C-Programm */
    printf("Hallo Welt!\n");
    exit(EXIT_SUCCESS);
}
\end{lstlisting}
\end{document}

```

Damit erhält man die folgende Ausgabe:

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    /* Ein simples C-Programm */
    printf("Hallo Welt!\n");
    exit(EXIT_SUCCESS);
}

```

Da sowohl die Schrift LuxiMono als auch die zugehörige L^AT_EX-Anpassung noch recht neu sind, wird man sie selbst installieren müssen, was aber nach den Angaben in [2] nicht schwer fällt. Anwender, die ihr T_EX-System von der neuesten T_EXLive-CD installiert haben, müssen sich darum nicht zu kümmern, da dort bereits alles Nötige enthalten ist.

Literatur

- [1] Carsten Heinz: *The Listings Package*; Apr. 2002; CTAN: [tex-archive/macros/latex/contrib/supported/listings/listings.dtx](#).
- [2] Walter Schmidt: *The luximono Package*; Juli 2002; CTAN: [tex-archive/fonts/LuxiMono/](#).

T_EX-Beiprogramm

EuroBachot_EX 2002 in Bachotek/Polen

Knut Lickert, Ulrik Vieth

Vom 29. April bis zum 3. Mai 2002 fand in Bachotek die 13. EuroT_EX, zugleich die 10. Bachot_EX statt. Dieser Bericht bietet eher einen (subjektiven) Eindruck über die Tagung; wer sich für die Vorträge interessiert, kann unter [1] nachsehen.

Samstag, 27. 4. und Sonntag, 28. 4.: Anfahrt

Bei der Entscheidung, nach Bachotek zu fahren oder nicht, spielt die Entfernung und der Aufwand der Reise eine große Rolle. Dank der NTG, die einen Bus gesponsort hat, der von Holland nach Polen fährt, fällt die Entscheidung schon leichter. So stehe ich Samstag abend kurz vor sieben Uhr mit Ulrik Vieth am Hauptbahnhof in Stuttgart und steige in den ICE nach Hannover, um dort auf den Bus der NTG zu wechseln.

In Hannover treffen wir dann an der S-Bahn zum Flughafen auf Martin Schröder, den dritten Zustieger in Hannover. Gemeinsam fahren wir zum Flughafen, wo der Bus schon auf uns wartet. Dort werden wir sogleich erwartungsvoll von Kim Roberts mit der Frage begrüßt, ob Martin den telefonisch bestellten Käse zum Rotwein nicht vergessen hat. Im Bus suchen wir uns dann zwischen den anderen T_EXies, deren Kindern und Hund einen Platz. Nach einer Stunde Fahrt ist langsam Ruhe eingekehrt und die meisten schlafen Berlin entgegen. Pech gehabt haben nur diejenigen, die den Platz neben dem Hund erwischt haben und die ganze Nacht durch lautes Hecheln wach gehalten werden.

Um drei Uhr nachts kommen wir am Flughafen Berlin-Schönefeld an. Wir vertreten uns die Beine und suchen eine Möglichkeit, etwas zu trinken, was einer der teuersten Kaffees unseres Lebens wurde. Kurz vor vier Uhr kommt

dann ein Anruf, wo der Bus bliebe: Am Bahnhof Schönefeld würden schon die Zusteiger aus Berlin warten. Auf den Hinweis, wir warten am Flughafen, kommt Volker Schaa innerhalb weniger Minuten und spielt den Lotsen zum nahe gelegenen Bahnhof, wo wir die letzten Mitfahrer aufnehmen.

Frühmorgens kommen wir an der deutsch-polnischen Grenze an. Unsere Pässe werden eingesammelt und den polnischen Grenzern übergeben. Ein belgischer Mitfahrer verzögert dann unsere Einreise nach Polen, weil er keinen Reisepass sondern nur einen Personalausweis dabei hat. Nach den üblichen Spielchen, dass er eigentlich nicht einreisen dürfte, wird eine Lösung gefunden. Die deutschen Grenzer stellen ihm einen provisorischen deutschen Pass aus, mit dem er dann nach Polen einreisen kann.

Die Fahrt durch Polen verläuft ereignislos; alle dösen, schlafen, tratschen oder schauen auf dem Bus-TV „Wallace and Grommit“. Nach einer Pause auf einer Raststätte und einem abgebrochenen Abstecher zu einem Geldautomaten in Torún (der Bus ist für die Brücke zu hoch), kommen wir um zwei Uhr nachmittags im Tagungsgelände an. Nach der allgemeinen Begrüßung werden wir auf die Hütten und Zimmer verteilt. Die Zeit bis zum Mittagessen um drei Uhr wird mit Duschen, Zimmer besiedeln und Sich-Umschauen verbracht. Nach dem Essen verschwinden die meisten wieder, um den während der Fahrt verpassten Schlaf nachzuholen. Einige machen sich auch zu einem Waldspaziergang auf, um die Gegend kennenzulernen.

Das Tagungsgelände [3] in Bachotek liegt idyllisch an einem wunderbaren kleinen See. Um den See hat es Wald und Wiesen. Unter den Bäumen sind verteilt auf dem Gelände Hütten verschiedenen Typs, in denen bis zu vier Leute untergebracht werden können. Die Anlage verführt sofort zu der Frage: Will ich mir hier wirklich im abgedunkelten Raum Vorträge anhören oder mich nicht mit einer Angel an den See setzen?

Um sieben Uhr gibt es Abendessen, danach versammeln sich verschiedene Gruppen, um den Vortragsraum zu inspizieren, den Rechnerraum zu begutachten und den Kiosk vor Ort zu testen. Ich setze mich ans Lagerfeuer, das Taco Hoekwater schon mit Begeisterung pflegt. Es wird viel geredet, Bier getrunken und sich regelmäßig gefragt, ob man wegen des Regens nicht lieber gehen sollte. Bevor die Entscheidung aber fällt, hört der Regen meistens wieder auf.

Montag, 29. 4.

Nach einer Nacht im Tiefschlaf geht es um acht Uhr morgens zum Frühstück und anschließend zur Eröffnung der Veranstaltung. Vor dem ersten Vortrag gibt es erst mal Unruhe, weil jeder sein Begrüßungspaket inspiziert. Neben den Tagungsunterlagen gibt es eine Tasche, einen Kaffeebecher, ein Mousepad und ein T-Shirt mit EuroBachTeX-Logo.

In der Kaffeepause kommt dann erst mal Arbeit auf mich zu: Volker fragt mich, ob ich nicht eine Kleinigkeit für „Die TeXnische Komödie“ schreiben will. Auf Martins Einwurf „Stell dir einfach vor, Luzia schaut dich fragend an“ gibt es keine Ausrede; das Ergebnis liegt hier vor.

Den Rest des Tages hat man dann die Wahl zwischen dem Besuch der Vorträge oder dem Schwänzen der Vorträge, wo man sich entweder im Rechenraum trifft oder sich zwischen den Bäumen einen Platz mit Seeblick sucht und die Sonne genießt. Ein besonderer Termin, den man nicht verpasst haben sollte, ist die Vorführung einer interaktiven CD-ROM über das Werk von Hermann Zapf, die Volker mitgebracht hat.

Nach dem Abendessen ist ein großes Lagerfeuer vorbereitet. Würstchen und Stöcke stehen bereit; es kann gegrillt werden. Dummerweise fängt es nach kurzer Zeit an zu regnen und es bleibt diesmal nicht bei einem kleinen Schauer, sondern es gibt einen heftigen Regenguss. Unter einem Sonnenschirm wird die mit Gitarren ausgerüstete polnische Schar schnell von einer dicht gedrängten Gruppe eingefasst. Es bilden sich Gitarrenquartetts und Chöre in wechselnder Besetzung, die für musikalische Unterhaltung sorgen. Bei jedem Regenguss verkleinert sich die Gruppe um das Lagerfeuer und die Gesangsgruppe unter dem Schirm wird größer. Hört der Regen auf, gibt es wieder eine Rückwanderung.

Der Regen vertreibt nur wenige, so wird es ein lustiger Abend mit diversen Bieren, vielen Gesprächen und viel Spaß. Das polnische Quartett erhält später mit David Kastrup noch deutsche Verstärkung. Spätabends regnet es dann auch nicht mehr und es gibt eine große Runde um das Lagerfeuer. Dank des Regens hat es auch kaum Mücken.

Dienstag, 30. 4.

Der Morgen begrüßt uns mit Sonnenschein. Für diejenigen, die wie am Vortag um acht Uhr frühstücken wollen, zeigt sich mal wieder: Ein gelegentlicher

Blick ins Tagungsprogramm lohnt sich. Eine halbe Stunde mehr Schlaf wäre die Belohnung gewesen.

Während des Tages gibt es wieder zahlreiche Vorträge. Zu den Vorträgen zum Thema „Open Source“ und `preview-latex` für Emacs gibt es lebhaftes Diskussionen, zum Thema Preview wird spät abends noch ein Workshop im kleinen Kreis einberufen. Nach dem Abendessen treffen sich die Polen zur GUST-Mitgliederversammlung; die Anderen verteilen sich in kleinen Gruppen auf dem Gelände.

Mit etwas Glück und Verbindungen bekommt man dann mit, in welcher Hütte es „abgeht“, nachdem die Bar schon um halb zwölf geschlossen hat. In Hütte 30 haben sich die Polen mit Wodka, leckerem Honig-Likör, Gitarren und Gesang versammelt. Je später der Abend, desto voller wird es. Irgendwann nach Mitternacht hat es fast vierzig Leute in der kleinen Hütte, jeder Fleck ist von jemand belegt. Die Überfüllung zwingt(?) manche, sich den Platz zu teilen und sich gegenseitig auf dem Schoß zu sitzen. So teilen sich Eric Frambach, Kim, Volker und Jerzy Ludwiczowski einen Platz in wechselnder Stapelfolge.

David beweist seine Tagungsqualitäten heute wieder sehr eindrucksvoll. Nachdem er gestern und heute schon insgesamt drei Stunden Vorträge gehalten hat, beweist er auch bei diesem Event seine Gesangsqualitäten. Ich gebe irgendwann in der Nacht auf und vernachlässige meine Chronistenpflicht, indem ich ins Bett gehe. Auf die Frage beim morgigen Frühstück, ob ich nachts auch in der Hütte gewesen sei, in der sie die Wände hochgeklettert sind, habe ich verneint. Inzwischen muss ich sagen: Ja, aber ich ging kurz bevor David auch noch seine Künste im Freiklettern unter Beweis stellte.

Mittwoch, 1. 5.

Diesen Morgen beim Frühstück sind deutlich weniger Leute anwesend als sonst. Erst zur Kaffeepause füllen sich die Reihen wieder. Ulrik wird von Kim als erstes mit der sorgenvollen Frage nach den Fotos der letzten Nacht begrüßt. Sie wird wohl nach den feucht-fröhlichen Feiern der Nacht nur noch geahnt haben können, dass da viele interessante Schnappschüsse von ihr gemacht worden sind, die wohl besser nicht veröffentlicht werden sollten (siehe [2]).

Die Vorträge vergehen im Fluge, Begeisterungstürme macht Marcin Woliński's Vortrag über „PowerPoint für Arme“, der mit einigen Tricks `TeX` zur Aus-

gabe einer Präsentation auf einer Textkonsole mit ANSI-Escape-Sequenzen benutzt.

Nach der Mittagspause geht es dann mit dem Bus in die Stadt Grudziądz (Graudenz). Wir werden gewarnt, dass der Bus um 15 Uhr auch ohne uns abfährt, wenn wir nicht pünktlich sind. Das bewirkt dann auch Rennerei zum Bus mit vom Duschen feuchten Haaren. Unterwegs geht dann auf einem Mobiltelefon ein Anruf eines Verspäteten ein, wir drehen also nochmal um. Wie jemand bemerkte, sollte die korrekte Aussage zur Pünktlichkeit wohl lauten: „we will depart without you, and then return to pick you up“.

Gegen 16 Uhr haben wir Grudziądz erreicht, wo es dann in vier Gruppen auf eine Stadtführung geht. Die deutsch-holländische Gruppe mit englischsprachiger Führung erleidet bald einen kleinen Schwund, als Taco und seine Freunde beschließen, in der ersten Kneipe am Marktplatz das Ende der Führung abzuwarten. Geschickterweise behalten sie auch gleich den Busfahrer bei sich, um die Rückfahrt nicht zu verpassen. Der Rest von uns folgt brav der Führung, die es sich nicht nehmen lässt, ausführlich zu erzählen, was alles zuerst von den Schweden im dreißigjährigen Krieg und später von den Deutschen im Zweiten Weltkrieg zerstört und dann wieder aufgebaut wurde. Zum Schluss erfahren wir, dass wir auch eine deutschsprachige Führung dabei hatten.

Nach knapp zwei Stunden sind schließlich alle Gruppen wieder am Treffpunkt angelangt. Nun geht es weiter ins Museum, wo Mitglieder der GUST eine Ausstellung über das Werk des Typographen Zygfryd Gardzielewski, dem Schöpfer der Schrift „Antykwa Torúnska“, vorbereitet haben, die an diesem Abend im Beisein der Presse und der Honoratioren der Stadt feierlich eröffnet wird.

Nach einer Stunde geht es dann weiter zum Tagungssessen in einer alten Mühle, die zum Ausflugslokal umgebaut wurde. Für uns ist der hintere Saal reserviert, wo bereits die Tische gedeckt sind und eine Blues-Band für gute Stimmung sorgt. Es dauert nicht lange, bis einige Paare auf der Tanzfläche zu finden sind.

Das Essen lässt derweil auf sich warten. Meist gelingt es nicht, alle Bestandteile eines Ganges gleichzeitig an einen Tisch zu bringen. So wird an einem Ende bereits der Salat verzehrt, während am anderen Ende noch auf die Suppe gewartet wird. Dann müssen mangels Salat die kalten Wurst- und Käseplatten herhalten, um die Lücke bis zum Hauptgang zu überbrücken. Später fehlt es

dann an Platz, um die vielen nicht abgeräumten Platten und Schüsseln auf dem Tisch zu stapeln. Trotzdem kann man über das Essen nicht klagen.

Nach dem Hauptgang folgen dann die Festreden, schließlich handelt es sich um die 10. BachTeX-Tagung. Bei dieser Gelegenheit wird auch bekanntgegeben, dass die GUST am Vortag neue Ehrenmitglieder gewählt hat. Neben der Schriftführerin Jolanta Szelatyńska wurden auch die langjährigen BachTeX-Gäste Volker RW Schaa (DANTE e.V.) und Eric Frambach (NTG) mit dieser besonderen Auszeichnung versehen. Nach den Ehrungen wird dann die Geburtstagstorte aufgeföhren, die anschließend an die Tagungsgäste verteilt wird.

Während einige noch mit Kaffee und einem zweiten Stück Torte beschäftigt sind, ist ein ständiges Kommen und Gehen an der Eingangstür zum Vorraum zu beobachten. Wie sich herausstellt, herrscht draußen im öffentlichen Teil des Lokals Party-Stimmung und es wird viel getanzt, während unsere Blues-Band gerade noch pausiert. Im Innenhof lodert ein Lagerfeuer und einige haben es sich auch dort gemütlich gemacht. Als die Blues-Band wieder zum Tanz aufspielt, kehren die meisten in den Saal zurück.

Auch an diesem Abend beweist David erneut seine vielseitigen Tagungsqualitäten, als er mit Jerzys Tochter einige sehr professionelle und rasante Tänze aufs Parkett legt. Gegen Ende des Abends wird die Blues-Band noch um einige Zugaben gebeten. Schließlich ist es dann aber doch Zeit zum Aufbruch.

Nachdem die ersten zwei Busse schon weg sind, verzögert sich die Abfahrt des dritten Busses fast um eine halbe Stunde, während ein holländischer Vater noch eines seiner Kinder sucht und telefonisch herausfinden lässt, ob in einem der anderen Busse vielleicht bei einer anderen Familien ein Kind zuviel mitgekommen ist. Schließlich löst sich die Sache in Wohlgefallen auf und wir können auch endlich die Heimfahrt antreten. Nach Ankunft gehen die meisten dann sofort ins Bett, schließlich ist an diesem Abend schon genug gefeiert worden.

Donnerstag, 2. 5.

Nach dem Frühstück gibt es zunächst polnische Vorträge, weshalb einige auch die Gelegenheit nutzen, etwas länger zu schlafen. Nach der Kaffeepause geht es mit Hans Hagen auf englisch weiter bis zum Mittag. Der Nachmittag ist dann wieder für polnische Vorträge reserviert.

Das Wetter ist besser geworden, sodass einige bereits zu einer Bootsfahrt auf dem See aufbrechen. Als Alternative wird kurzfristig eine kurze Einkaufstour mit dem Bus ins nahe gelegene Brodnica organisiert, um Vorräte für die Rückfahrt einkaufen zu können. Gegen Abend berichten Martin und Stacjek dann noch kurz über den Stand der T_EXLive 7, deren letzte Tests während der Tagung durchgeführt wurden.

Nach dem Abendessen ist ein weiteres Lagerfeuer angesetzt worden. Im Gegensatz zu Montag, wo es kühl und regnerisch war, bleibt es an diesem Abend trocken und warm bis spät in die Nacht. Der Rauch des Feuers hilft ein wenig, die Mücken zu vertreiben, die an diesem Tag schon recht unangenehm sind. Erneut werden Würstchen über dem Feuer gegrillt, Bier getrunken und viel geredet. Zwischendurch gibt es von Taco und seinen Freunden eine Vorführung im Feuerspucken. Auch Kim wird mit Erfolg in diese Kunst eingewiesen. Dass an diesem Abend auch wieder musiziert und gesungen wurde, versteht sich fast schon von selbst. So wird es eine lange und ruhige Nacht in der freien Natur bei sternklarem Himmel. Auch als um ein Uhr nachts das Bier alle ist, bleiben noch viele sitzen. Eine Gruppe geht gegen drei Uhr ins Bett, am längsten bleiben die Polen. Wie beim Frühstück zu hören war, haben einige bis fünf Uhr ausgehalten. Anschließend muss es dann schon hell geworden sein.

Freitag, 3. 5.

Trotz etwas wenig Schlaf heißt es zeitig aufstehen, denn auf dem Programm steht der gemeinsame Vortrag von Yannis Haralambous und John Plaice über den Stand von Omega. Anschließend geht es mit polnischen Vorträgen weiter, weshalb die meisten Westeuropäer es vorziehen, sich draußen aufzuhalten. Auf der Terrasse vor dem Kiosk gibt es auf einem Notebook eine erneute Vorführung der Zapf-CD-ROM für diejenigen, die am Montag nicht dabei waren. Die Tschechen und Ungarn brechen bereits zum Heimweg auf und es gibt herzliche Abschiedsszenen.

Nach dem Mittagessen setzen sich dann viele bei herrlichem Wetter in die Sonne oder gehen sogar baden – übrigens an einem Tag, an dem es in Deutschland noch nasskalt und regnerisch war. Erst zur Abschlussveranstaltung am Spätnachmittag kehren die meisten noch einmal in den Vortragsraum zurück, wo für die TUG-Tagung 2002 in Kerala und 2003 in Hawaii sowie die EuroT_EX-Tagung 2003 in Brest (Frankreich) geworben wird. Nach dem offiziellen Schluss der Veranstaltung bleiben dann nur noch ein paar Stunden

bis der Bus fährt, um die letzten Sonnenstrahlen zu genießen, seine Sachen zu packen und das letzte polnische Bier zu konsumieren.

Nach dem Abendessen versammeln sich alle gegen halb acht auf der Wiese neben dem Busparkplatz. Die einen, um ihr Gepäck zu verstauen und einen Sitzplatz zu reservieren, die anderen, um herzlich Abschied zu nehmen, wie es in Polen üblich ist. Zurück bleiben am letzten Abend nur noch die Einheimischen und einige individuell angereiste Teilnehmer aus Westeuropa, die erst am Samstag ab Warschau zurückfliegen.

Freitag, 3. 5. und Samstag, 4. 5.: Rückfahrt

Pünktlich um acht Uhr abends macht sich der EuroTeX-Bus auf die Rückfahrt. Die Fahrt verläuft ruhig und ohne Zwischenfälle, sodass bald geschlafen werden kann. Nach einer Pause auf einer Raststätte um Mitternacht erreichen wir gegen 2 Uhr 30 die Grenze, wo uns allerdings ein Rückstau von sechs Bussen erwartet, sodass wir erst um 4 Uhr 50 weiterfahren können. Gegen 6 Uhr 30 erreichen wir Berlin-Tegel, wo uns die Dänen und Amerikaner verlassen. Gegen 9 Uhr 30 erreichen wir dann den Flughafen Hannover, wo bis auf David sämtliche deutschen Teilnehmer aussteigen und die Ablösung für den Busfahrer für den Rest der Strecke nach Utrecht bereits wartet. Wir verabschieden uns von den im Bus verbleibenden („Bis nächstes Jahr“) und überreichen dem Busfahrer als Dank noch einen TeX-Löwen.

język polski	english	deutsch
Dzień dobry	good morning	Guten Morgen
Kolacja	dinner	Abendessen
Obiad	lunch	Mittagessen
Śniadanie	breakfast	Frühstück
○	L	D
▽	G	H
dziękuję	thank you	Danke
piwo	beer	Bier
tak	yes	ja

Tabelle 1: Polnisch für Anfänger

Wer Lust bekommen hat, die internationale T_EX-Gemeinde persönlich kennen zu lernen: Die BachoT_EX-Tagung findet jedes Jahr um den ersten Mai statt. Die wichtigsten polnischen Worte sind in Tabelle 1 enthalten.

Für Reisen in andere fernere Gegenden empfiehlt sich die TUG 2002-Tagung. Sie findet vom 4. bis 7. September 2002 in Kerala (Indien) statt, die nächste EuroT_EX-Tagung vom 24. bis 27. Juni 2003 in Brest (Frankreich), gefolgt von der TUG 2003 auf Hawaii vom 20. bis 24. Juli 2003.

Literatur

- [1] *EuroBachoT_EX 2002 Program*; <http://www.gust.org.pl/EuroBachoTeX/program.html>.
- [2] *Photos der EuroBachoT_EX*; <http://www.gust.org.pl/EuroBachoTeX/photos.html>.
- [3] *Plan of the Bachotek Holiday Retreat*; http://www.gust.org.pl/EuroBachoTeX/plan_bacho.html.

Tipps und Tricks: PDF-Datei vorhanden – Broschüre gewünscht

Rolf Niepraschk

Ein Kollege bekam eine Digitalkamera zum Geburtstag geschenkt. Wie immer häufiger anzutreffen, lag nur eine sehr kurze Betriebsanleitung bei, die an interessanten Stellen auf eine CD-ROM verwies. Diese enthielt eine ausführliche 150-seitige Beschreibung als PDF-Datei. Sehr verständlich war sein Wunsch, die Datei auszudrucken und zwar möglichst sparsam und in einer Art, dass man später daraus leicht eine Broschüre machen kann. Ich sollte helfen. Zuerst fielen mir die Programme `psbook` und `pstops` ein. Sie wären wohl nach Wandlung der PDF-Datei ins PostScript-Format brauchbar. Dann aber erinnerte ich mich an eine andere Möglichkeit, die ich im Folgenden vorstellen will. Es geht auch mit T_EX, genauer: mit pdfL^AT_EX. Das L^AT_EX-Paket

pdfpages bietet alles Notwendige. Nach dem Lesen der Dokumentation [1] und etwas Probieren ergab sich die folgende erstaunlich einfache Lösung:

```
\documentclass[a4paper]{article}% 'DI_7_5.tex'
\usepackage{pdfpages}[2002/04/23]
\begin{document}
  \includepdf[pages=-,offset=0cm 0cm, delta=0cm 1cm,scale=.9,
             signature=2,landscape]{DI_7_5_DE_HW.pdf}
\end{document}
```

Kurz zur Erklärung: Die Anweisung `\includepdf` fügt Seiten einer PDF-Datei (der notwendige Parameter) in ein \LaTeX -Dokument ein. Die Angabe `pages=-` wählt sämtliche Seiten aus; `offset` und `delta` geben die Verschiebung der Einzelseiten bzw. den Abstand zwischen ihnen an. Besonders nützlich ist der Parameter `signature`, da er die Logik zur Anordnung der Seiten für das spätere Zusammenstellen der Broschüre bietet. Mit `scale` kann die Größe der eingefügten Seiten verändert werden, wie von der Anweisung `\includegraphics` bekannt. Nähere Hinweise zu den Parametern enthält die Dokumentation des Pakets. Nach Aufruf von

```
pdflatex DI_7_5.tex
```

entstand aus der Datei mit 150 kleinen Seiten (jede ca. 15 cm lang und breit) eine neue PDF-Datei mit 76 A4-Seiten. Auf jeder Seite befinden sich zwei der kleinen Originale nach folgendem Schema:

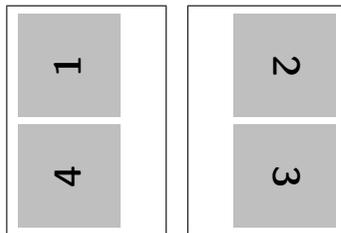


Abbildung 1: Vorder- und Rückseite des ersten A4-Blattes

Die Rückseiten sind jeweils um 180° gedreht, sodass zusammengehörende Teilseiten nach Ausdruck auf einem doppelseitig druckenden Drucker gleichsinnig auf das Papier kommen. Zwischen den beiden Teilseiten wurde ein zusätz-

licher Abstand von 1 cm eingefügt, um die später in der Hälfte gefalteten Blätter leichter zusammenheften zu können. Mein Kollege war zufrieden.

Literatur

- [1] Andreas Matthias: *The pdfpages Package*; Apr. 2002; CTAN: tex-archive/macros/latex/contrib/supported/pdfpages/pdfpages.dtx.

Tipps und Tricks: Kartenerstellung mit PanMap

Rainer Sieger

In der Zeitschrift „Die T_EXnische Komödie“ 2/2002 erschien in der Rubrik Tipps und Tricks ein Artikel von Gerd Neugebauer über die Erstellung von Illustrationen für L^AT_EX. Als Beispiel wurde die Entwicklung einer Landkarte dargestellt. Mit Hilfe des Programms PanMap kann der Teil der reinen Kartenerstellung erheblich vereinfacht werden.

Einleitung

Das Programm PanMap wurde im Rahmen des Projektes PANGAEA entwickelt, um auf einfache und schnelle Weise geocodierte Daten in Karten darstellen zu können. Das Programm wird für die Betriebssysteme MacOS und Windows über den PANGAEA Web-Server, zusammen mit verschiedenen geografischen Ressourcen im PanMap-Format, als Freeware verteilt [1].

PanMap stellt georeferenzierte Daten, das heißt Informationen, die einen geografischen Bezug besitzen, in einer Karte dar. Es kann also als eine Art Mini-GIS (GIS steht für Geographisches Informations-System) betrachtet werden.

Dabei stehen einige gängige kartografische Projektionen zur Verfügung. Kartenausschnitt und Maßstab sind interaktiv wählbar, die grafische Darstellung importierter Daten und das Layout einer Karte können vom Benutzer definiert werden. PanMap arbeitet vektororientiert; Rasterdaten können mit

Tabelle 1: Positionen der T_EX-Stammtische

<i>Latitude</i>	<i>Longitude</i>	<i>Stadt</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Stadt</i>
53.5611	10.0181	Hamburg	50.8369	12.9383	Chemnitz
53.0756	8.825	Bremen	49.5833	11.0136	Erlangen
52.3706	9.7292	Hannover	49.4019	8.6822	Heidelberg
52.5757	13.2931	Berlin	49.0039	8.3564	Karlsruhe
51.9697	7.6539	Münster	48.7758	9.1825	Stuttgart
51.5058	7.5	Dortmund	48.1317	11.5497	München
51.2675	7.2017	Wuppertal	47.9983	7.8219	Freiburg
50.9444	7.0139	Köln	47.67	9.1731	Konstanz
51.0656	13.7306	Dresden			

PanMap nicht dargestellt werden. Außerdem können leider die entstehenden Flächen nicht eingefärbt werden.

Georeferenzierte Datensätze werden in ein PanMap-eigenes Format gewandelt und als Layer vorgehalten und benutzt. Jeder einzelne Layer kann individuell bearbeitet, gestaltet und in unterschiedlichen Karten verwendet werden. Gemeinsam mit einer Reihe von zusätzlichen Eigenschaften bilden ein oder mehrere Layer eine PanMap-Karte. Zu den Eigenschaften gehören beispielsweise der Kartenausschnitt, die geografische Projektion, Maßstab und Layout.

Die Karte der T_EX-Stammtische

Als Beispiel sei hier nochmals die von Gerd Neugebauer dargestellte Karte der T_EX-Stammtische nachgebaut. Neben dem eigentlichen Programm benötigen wir einen Layer für den Umriss von Deutschland. Zu finden ist dieser im Paket `Countries` [2]. Für die Positionen der Städte kann man den Times-Atlas oder das Internet befragen oder man benutzt das Paket `Cities` [3]. Aus den vielen angezeigten Städten schreibt man sich die Positionen der gewünschten heraus und fasst sie in einer Import-Tabelle zusammen. Diese (ASCII-)Tabelle beinhaltet nur drei Spalten. In der ersten steht die Latitude (Breite), in der zweiten die Longitude (Länge) und in der dritten der Name der Stadt (siehe Tabelle 1). Vorsicht! PanMap versteht als Dezimaltrenner nur den Punkt.

Diese Tabelle muss anschließend einmalig in PanMap importiert werden und steht dann als Layer zur Verfügung. Das eigentliche Erstellen der Karte geht nun schnell von der Hand und ist im Manual ausführlich erklärt [4].

Die mit PanMap erstellte Karte muss vor der Verwendung mit L^AT_EX in das eps- bzw. PDF-Format gebracht werden. Dazu kann man die Karte mit einem PostScript-Treiber in eine Datei drucken und dann mit Ghostscript weiterverarbeiten. Mir ist es jedoch nicht gelungen, den etwas großen Rand zu beschneiden und die Karte zu skalieren. Ich habe daher die Karte per Kopieren und Einfügen in ein vektorfähiges Grafikprogramm übernommen und dessen Exportfunktion benutzt. Diesen Weg wird man in der Regel immer anwenden müssen, da die Grafik nur so mit zusätzlichen Informationen versehen werden kann.

Die fertige Karte (siehe Abbildung 1) bindet man dann wie von Gerd Neugebauer vorgeschlagen mittels `\includegraphics` in das Dokument ein.

Literatur

- [1] <http://www.pangaea.de/Software/PanMap>.
- [2] <http://www.pangaea.de/Software/PanMap/Layers/Countries.zip>.
- [3] http://www.pangaea.de/Software/PanMap/Cities/Cities_CIA.zip.
- [4] Michael Diepenbroek, Hannes Grobe und Rainer Sieger: *PanMap*; 2000; http://www.pangaea.de/Software/PanMap/PanMap_Manual_de.pdf.

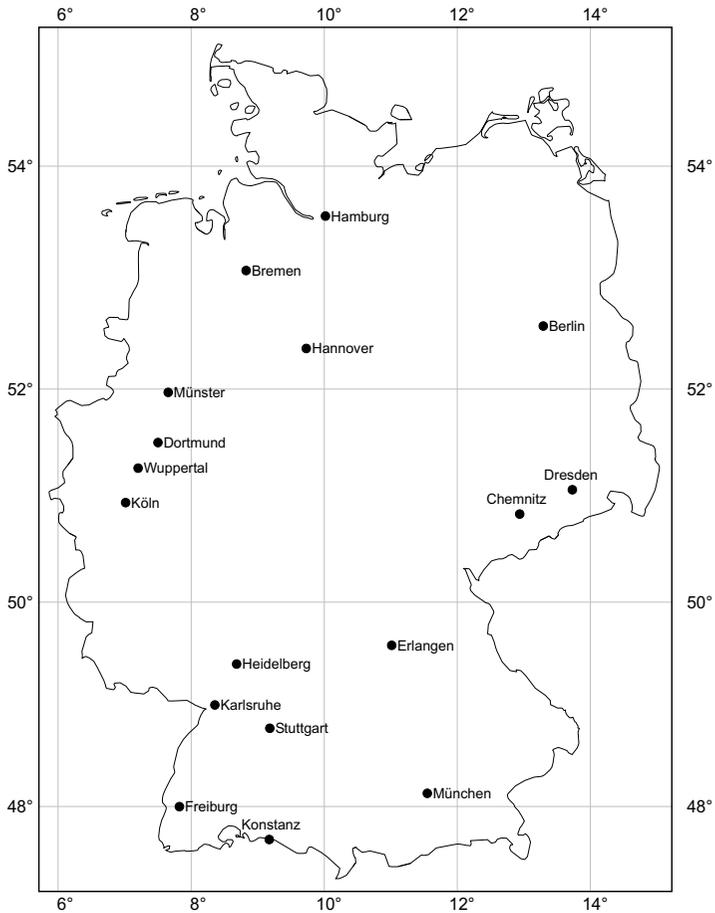


Abbildung 1: Die mit PanMap erstellte Karte.

Spielplan

Termine

- 4.–5. 10. 2002** Herbsttagung und 27. Mitgliederversammlung von DANTE e.V., Augsburg
<http://www.dante.de/dante/events/mv27/>
- 9.–14. 10. 2002** Frankfurter Buchmesse 2002
<http://www.frankfurter-buchmesse.de/>
Kontakt: Buchmesse Frankfurt
- 12. 10. 2002** Herbst-Treffen und Mitgliederversammlung der UK-TUG
Department of Computing Science and Information Technology
Nottingham University (Jubilee Campus), Nottingham, Großbritannien
- 20.–23. 10. 2002** CIDE 2002 – 5e Colloque International sur le Document Électronique
Hammamet, Tunesien
Kontakt: Conférence CIDE 2002
- 2.–4. 4. 2003** DANTE 2003 und 28. Mitgliederversammlung von DANTE e.V.
Universität Bremen, Bremen
Kontakt: DANTE e.V.
- 24.–27. 6. 2003** EuroT_EX 2003
14th Annual Meeting of the European T_EX Users' Group
Bretagne, Frankreich
Kontakt: Secrétariat de la conférence EuroT_EX 2003
- 20.–26. 7. 2003** TUG 2003 Waikaloa Beach Resort, Big Island, Hawaii
<http://www.tug.org/tug2003/>
Kontakt: Wendy McKay

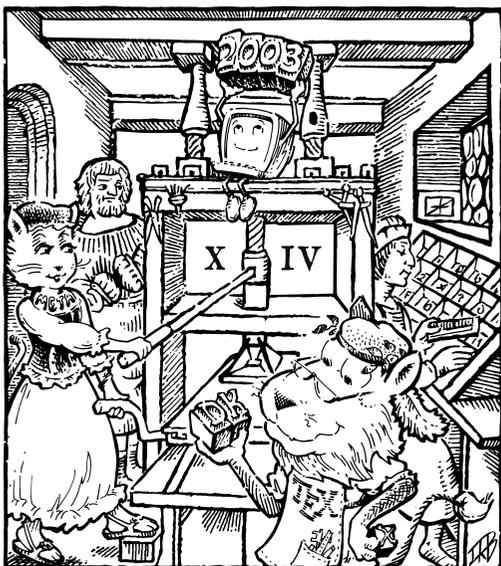
14th European T_EX Conference

ELECTRONIC DOCUMENT ☞ DIGITAL TYPOGRAPHY ☞ TYPOGRAPHICAL HERITAGE AND ITS SURVIVAL IN THE COMPUTER ERA ☞ NEW TECHNOLOGIES APPLIED TO EDUCATION ☞ BOOK DESIGN AND PRODUCTION PROCESS ☞ FONTS ☞ DOCUMENT INTERNATIONALIZATION AND EXCHANGE

Deadline for abstract submission: January 7, 2003

Organizer: Yannis HARALAMBOUS, École Nationale Supérieure des Télécommunications de Bretagne

Back to Typography



June 24-27, 2003
ENST de Bretagne
Brest 🇫🇷 Brittany
France

Programme Committee:

Jacques ANDRÉ (IRISA)
Hans HAGEN (PRAGMA)
Yannis HARALAMBOUS (ENST-BRETAGNE)
Alan HOENIG (CUNY)
Ronan KERYELL (ENST-BRETAGNE)
Frank MITTELBACH (EDS)
John PLAICE (UNSW)
Sebastian RAHTZ (OUS)
David SALOMON (CSU)
Petr SOJKA (MUB)

Contact:

<http://omega.enstb.org/eurotex2003>
<http://www.enst-bretagne.fr>



Die T_EXnische Komödie 3/2002

Invitation to TUG 2003
Silver Anniversary—25 Years!—of T_EX



ALOHA! T_EX enthusiasts!

Please join us in Hawaii for TUG 2003, the 24th Annual Meeting and Conference, where we will celebrate the Silver Anniversary—25 years!—of T_EX with leis, luaus, and laughter.

Did you know that “Aloha” means both “goodbye” and “hello”? We’ve already said goodbye to the 20th century, and the challenges of the 21st are upon us.

A retrospective of T_EX’s 25-year history is planned, as well as presentations and workshops on T_EX’s role in the 21st century.

Where do you see T_EX going? Where do you wish, hope, pray it will go? Where have you taken T_EX?

Grab your sunglasses and your sunscreen and head for the Outrigger Waikoloa Beach Resort, Big Island, Hawaii! Add your voice to our celebrations and discussions from July 20–24, 2003.

Professor Hermann Zapf, renowned font designer and calligrapher, and Duane Bibby, famous cartoon artist of T_EX and Meta lion drawings, will be our honored guests. Duane will have his pen ready to autograph your shirts, books, and anything you have with the T_EX Lion on it, when he is not snorkeling in the blue Hawaiian waters nearby.

Please register at the *earliest* possible opportunity at the TUG 2003 website:

<http://www.tug.org/tug2003>

so that we can plan well ahead to have a truly grand celebration of our accomplishments with T_EX.

We invite those of you who are developing new macros, methods, and management systems involving T_EX or T_EX Fonts to submit an abstract for a talk in answer to our “Call for Abstracts”. See

<http://www.tug.org/tug2003/callfor.html>

We also invite those of you who have been using T_EX since its birth in 1978, to contact us if you have interesting tales or anecdotes to share. The T_EX Heritage Committee, chaired by Sebastian Rahtz, will be compiling a celebratory volume for the Silver Anniversary of T_EX. The volume will consist of articles on reflections by key developers and users of T_EX and will be (temporarily) entitled “*T_EX Retrospective: milestones and byways on the road to beautiful typography*”

Also, write and tell us what topics you may be interested in learning about through Panel Discussions, Workshops, or Birds-of-a-Feather Sessions at the conference.

Important Dates To Remember

2002, Nov 18	Abstracts Due
2002, Dec 18	Abstracts accepted
2002, Dec 31	Early-Lion Registration Deadline (discount rate)
2003, Feb 9	First Draft Paper Due
2003, Apr 9	Registration Deadline
after 2003, Apr 9	Late registration (at additional cost)
2003, Jun 9	Final Paper Due
2003, Jul 1	Slides Due
2003, Jul 20	Conference Begins
2003, Jul 24	Conference Ends

Keep checking the website for updates and/or join the TUG2003 participants mailing list for the latest news and developments if you are planning to come.

Mark your calendars now, and start making plans to come to Hawaii. We hope to bid “Aloha!” to lots of old, familiar faces and lots of fresh, new ones. You don’t want to miss this one!

TUG 2003 CONFERENCE ORGANIZERS

T_EX Users Group

tug2003@tug.org

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-Anwendern statt, die für jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://www.dante.de/events/stammtische/>.

Berlin

Rolf Niepraschk
Tel.: 030/348 13 16
niepraschk@ptb.de
*Gaststätte „Bärenschenke“
Friedrichstr. 124
Zweiter Donnerstag im Monat, 19.00 Uhr*

Bremen

Martin Schröder
Tel.: 04 21/2 23 94 25
martin@oneiros.de
*Wechselnder Ort
Erster Donnerstag im Monat, 18.30 Uhr*

Chemnitz

Ralf König
Tel.: 03 71/5 90 54 75
ralf.koenig@s1998.tu-chemnitz.de
*Universitätsteil 1, Straße der Nationen 62,
Raum 1/068
Dritter Mittwoch im Monat, 18.00 Uhr*

Dortmund

Stephan Lehmke
Stephan.Lehmke@cs.uni-dortmund.de
*Café Durchblick
Universität Dortmund, Campus Nord
Zweiter Mittwoch im Monat, 20.00 Uhr*

Dresden

Hilmar Preuße
hille42@gmx.de
*Medien- und Kulturzentrum Pentacon,
Schandauer Str. 64
Erster Mittwoch im Monat, 19.00 Uhr*

Erlangen

Walter Schmidt, Peter Seitz
was@VR-Web.de,
*Gaststätte „Erlanger Gärtla“
Marquardsenstraße 1
Dritter Dienstag im Monat, 19.00 Uhr*

Freiburg

Heiko Oberdiek
Tel.: 07 61/434 05
oberdiek@uni-freiburg.de
*Gaststätte „Aquila“
Sautierstr. 19
Dritter Donnerstag im Monat, 19.30 Uhr*

Hamburg

Volker Hüttenrauch
volker_huettenrauch@hh.maus.de
*Vereinsheim der Hamburger
Microcomputer-Hochschulgruppe
Grindelallee 143 (Hinterhof)
Letzter Donnerstag im Monat, 18.00 Uhr*

Hannover

Mark Heisterkamp
heisterkamp@rrzn.uni-hannover.de
*Seminarraum RRZN
Schloßwender Str. 5
Zweiter Mittwoch von geraden Monaten,
18.30 Uhr*

Heidelberg

Luzia Dietsche
Tel.: 062 21/54 45 27
luzia.dietsche@urz.uni-heidelberg.de
*China-Restaurant „Palast“
Lessingstr. 36
Letzter Mittwoch im Monat, 20.00 Uhr*

Karlsruhe

Klaus Braune
Tel.: 07 21/6 08 40 31
braune@rz.uni-karlsruhe.de
*Universität Karlsruhe, Rechenzentrum
Zirkel 2, 3. OG, Raum 316
Erster Donnerstag im Monat, 19.30 Uhr*

Köln

Bruno Hopp

b.hopp@lepkes-frings.de

*Institut für Kristallographie**Zülpicher Str. 49b**Zweiter Mittwoch im Monat, 19.30 Uhr***Konstanz**

Matthias Weisgerber, Hraban Ramm

weisgerb@fmi.uni-konstanz.de,

hraban@fief.net

*Restaurant Rheingold**Spanierstrasse 3**Zweiter Donnerstag im Monat, 19.00 Uhr***München**

Michael Niedermair

m.g.n@gmx.de

*Gastwirtschaft „Rhætenhaus“**Luisenstr. 27**Erster Dienstag im Monat, 19.00 Uhr***Münster**

Johannes Reese

reesej@uni-muenster.de

*Gaststätte „Sabroso“**Mauritzstr. 19**Erster Montag im Monat, 20.00 Uhr***Stuttgart**

Bernd Raichle

bernd.raichle@gmx.de

*Gaststätte „Alte Mira“**Büchsenstr. 24**Zweiter Dienstag im Monat, 19.30 Uhr***Wuppertal**

Andreas Schrell

Tel.: 02 02/50 63 81

schrell@wupperonline.de

*Restaurant Croatia „Haus Johannisberg“**Südstr. 10**an der Schwimmoper Wuppertal-Elberfeld**Zweiter Donnerstag im Monat, 19.30 Uhr*

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10⁰⁰–12⁰⁰ Uhr)
Fax: 0 62 21/16 79 06
E-Mail: dante@dante.de

Konten: Volksbank Rhein-Neckar eG
BLZ 670 900 00
Kontonummer 2 310 007
Postbank Karlsruhe (Auslandsüberweisungen)
BLZ 660 100 75
Kontonummer 213 400 757

Präsidium

Präsident:	Volker RW Schaa	president@dante.de
Vizepräsident:	Klaus Höppner	vice-president@dante.de
Schatzmeister:	Tobias Sterzl	treasurer@dante.de
Schriftführer:	Günter Partosch	secretary@dante.de
Beisitzer:	Thomas Koch	
	Bernd Raichle	advisor@dante.de

Server

ftp: [ftp.dante.de](ftp:dante.de) [134.100.9.51]
E-Mail: ftpmail@dante.de
WWW: <http://www.dante.de/>

Autoren/Organisatoren

- Florian Benischke** [40] **Gerd Neugebauer** [3]
 Abteilung für Vegetationsökologie
 Althanstraße 14
 A-1091 Wien
 Im Lerchelsböhl 5
 64521 Groß-Gerau
 gene@gerd-neugebauer.de
- Ausstellungs- und Messe GmbH** [59] **Rolf Niepraschk** [31, 43, 53]
Buchmesse Frankfurt
 Reineckstr. 3
 D-60313 Frankfurt am Main
 visitor@book-fair.com
 niepraschk@ptb.de
- Conference CIDE 2002** [59] **Volker RW Schaa** [4]
 Département d'informatique
 Campus II, Bd Maréchal Juin
 Université de Caen
 F-14032 Caen Cedex
 France
 siehe Seite 66
 cide2002@infodoc.unicaen.fr
- Secrétariat de la conférence EuroTeX 2003** [59] **Dr. Rainer Sieger** [55]
 Département Informatique
 ENST Bretagne
 France
 Zum Mühlacker 9
 27619 Schiffdorf
 Yannis.Haralambous@enst-bretagne.fr
- Rudolf Herrmann** [7] **Karel Skoupy** [19]
 R. Alves de Sousa n° 1, 4° D
 P-2745-236 Queluz
 Portugal
 Computer Systems Institute
 ETH Zentrum, RZ H22
 CH-8092 Zürich
 skoupy@inf.ethz.ch
- Klaus Höppner** [4] **Ulrik Vieth** [45]
 siehe Seite 66
 Vaihinger Str. 69
 D-70567 Stuttgart
 ulrik.vieth@tesionmail.de
- Knut Lickert** [45] **Herbert Voß** [40]
 Franziskanergasse 15
 73728 Esslingen
 knut@lickert.net
 Wasgenstr. 21
 14129 Berlin
 voss@perce.de
- Wendy McKay** [59] **Dr. Roland Weibezahn** [5]
 wgm@cds.caltech.edu
 DANTE 2003
 Universität Bremen
 Institut für Wissenschaftliche Datenverarbeitung
 Postfach 330 440
 28334 Bremen
 dante2003@dante.de

Die T_EXnische Komödie

14. Jahrgang Heft 3/2002 Oktober 2002

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 5 T_EX-Tagung DANTE 2003 in Bremen –
Einladung und Call for Papers

Bretter, die die Welt bedeuten

- 7 Poster im Format DIN A0 mit `a0poster` und `textpos`
- 19 Extending $\mathcal{N}\mathcal{T}\mathcal{S}$ (by $\mathbf{x}\mathbf{T}\mathbf{S}$)
- 31 Tipps und Tricks: Tierkunde einmal anders
- 34 `gauss.sty` – Visualisierung des Gaußschen Eliminationsverfahrens
- 41 Hüllen für Musikkassetten
- 43 Tipps und Tricks: Fette Schreibmaschinenschrift

T_EX-Beiprogramm

- 45 EuroBachoT_EX 2002 in Bachotek/Polen
- 53 Tipps und Tricks: PDF-Datei vorhanden – Broschüre gewünscht
- 55 Tipps und Tricks: Kartenerstellung mit PanMap

Spielplan

- 59 Termine
- 60 EuroT_EX 2003
- 61 Invitation to TUG 2003 Silver Anniversary — 25 Years! — of T_EX
- 64 Stammtische

Adressen

- 67 Autoren/Organisatoren