

# Die T<sub>E</sub>Xnische Komödie

---

DANTE  
Deutschsprachige  
Anwendervereinigung T<sub>E</sub>X e.V.

14. Jahrgang Heft 4/2002 Dezember 2002

4/2002

# Impressum

---

„Die T<sub>E</sub>Xnische 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<sup>A</sup>T<sub>E</sub>X-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L<sup>A</sup>T<sub>E</sub>X-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<sub>E</sub>X e.V.  
Postfach 10 18 40  
69008 Heidelberg

E-Mail: [dante@dante.de](mailto:dante@dante.de)

[dtk-redaktion@dante.de](mailto: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 1/2003: 10. Januar 2003

ISSN 1434-5897

# Editorial

---

Liebe Leserinnen und Leser,

Das Jahr geht zu Ende und damit liegt jetzt auch die letzte Ausgabe der Mitgliederzeitschrift dieses Jahrgangs vor. Wie immer nach Mitgliederversammlungen ist auch in dieser Ausgabe einiger Platz dem Vereinsinternen gewidmet. In der Vergangenheit war es dann schon einmal so, dass daneben nicht viel  $\TeX$ nisches oder Typographisches zu lesen war. Dies ist in dieser Ausgabe ganz anders. Ich glaube, ich erspare es mir, die einzelnen Beiträge noch einmal anzupreisen. Das Inhaltsverzeichnis, die Autoren und die Beiträge sprechen für sich selbst.

Auf einen  $\TeX$ nischen Punkt möchte ich an dieser Stelle ganz besonders eingehen, da er ansonsten nicht zu sehen ist. Bei der Korrektur der Artikel wollte ich eine Kleinigkeit an einer Abbildung in dem Beitrag über Satzspiegelkonstruktionen geändert haben – die Schrift vor den Linien anstatt dahinter. Normalerweise liegen Abbildungen als EPS- oder PDF-Dateien vor. Je nachdem, mit welchem Verfahren diese erzeugt worden sind, ist es dann meistens das einfachste, den Autoren um eine Änderung zu bitten. Bei diesem Beitrag waren aber keine Abbildungsdateien dabei. Ein Blick in die Quellen hat dann auch gezeigt, dass die Abbildungen gänzlich mit  $\LaTeX$ -Mitteln gemacht worden sind. Das zeigt, dass selbst dieses Mittel noch lebt und dass man damit brauchbare Ergebnisse erzielen kann. Und damit war es für mich dann auch ein Leichtes, meine gewünschten Änderungen selbst vorzunehmen.

Ich wünsche am Schluss allen Lesern ein frohes Weihnachtsfest und einen guten Rutsch ins neue Jahr, nicht ohne den Wunsch auszusprechen, dass sich in dieser geruh-samen Zeit der Eine oder die Andere das durchringt und einen – mehr oder weniger – kleinen Beitrag für „Die  $\TeX$ nische Komödie“ zu schreiben. Damit kann dann hoffentlich das erste Heft im nächsten Jahr so interessant werden, wie es das letzte in diesem Jahr ist.

Mit  $\TeX$ nischen Grüßen

Ihr Gerd Neugebauer

# Hinter der Bühne

---

## Vereinsinternes

### Grußwort

Liebe Mitglieder,

dieses Grußwort bricht mit der Tradition – so man nach vier Jahren davon sprechen mag –, von beiden Vorsitzenden gemeinsam zu stammen. Der Grund hierfür ist allerdings harmlos: Volker RW Schaa hat sich rechtzeitig vor Redaktionsschluss in den Urlaub verabschiedet. Ich werde diese Gelegenheit prompt nutzen und in der zweiten Hälfte dieses Grußwortes eher persönliche Gedanken ausbreiten.

Beginnen möchte ich mit einer Rückschau auf das Herbsttreffen in Augsburg. Es war eine sehr angenehme Tagung, nicht zuletzt wegen der hervorragenden Arbeit, die Gerhard Wilhelms und seine Helfer geleistet haben. Einige Ideen (z. B. die Tafel mit den Klebepunkten) werden wir in der Zukunft bestimmt aufgreifen. Persönlich gefreut haben mich die Tutorien von Personen, deren Gesichter noch nicht seit Jahren allgemein bekannt sind, und die große Teilnehmerzahl, die mit über 60 praktisch das Niveau von Frühjahrstagungen erreichte.

Nach den Terminen für die beiden DANTE-Tagungen im nächsten Jahr, die schon im letzten Grußwort Thema waren, stehen auch die Termine der Euro $\TeX$  2003 und TUG 2003 fest. Die Euro $\TeX$  2003 wird von Yannis Haralambous vom 24. bis 27. Juni 2003 an der Hochschule für Telekommunikation ENST ausgerichtet. Diese befindet sich etwas außerhalb von Brest in Frankreich. Wie der Zufall es will, nahm ich an genau diesem Ort vor zehn Jahren an einer physikalischen Konferenz teil. Ich bin sicher, dass die reizvolle bretonische Küste zu einer angenehmen Atmosphäre beitragen wird (nach einer unruhigen Bootsfahrt zur Insel Ouessant damals empfehle ich allerdings, an Land zu bleiben). Exotisch, aber für europäische Teilnehmer leider schwer erschwinglich, ist der Veranstaltungsort der TUG 2003. Diese findet vom 20. bis 24. Juli 2003 in Hawaii statt.

Eine Frage, die im letzten Grußwort noch offen bleiben musste, ist inzwischen geklärt. Der diesjährige CTAN-Abzug ist rechtzeitig fertig geworden. Wie Sie sicher gemerkt haben, ist er auf vier CD-ROMs angewachsen. Diese Entscheidung wurde kurzfristig getroffen, nachdem klar wurde, dass ein Zusammenstauchen des Inhalts von CTAN auf drei CD-ROMs nur mit Mühe und Schmerzen zu machen gewesen wäre. Möglich wurde dieser Schritt dank der gewohnt guten Zusammenarbeit mit der Fachbuchhandlung Lehmanns. Es stellt sich die Frage – wie bereits auf der Mitgliederversammlung in Augsburg andiskutiert – nach der weiteren Zukunft dieses jährlichen Projektes. Die meiste Arbeit, die ich als bisheriger Maintainer des Abzugs investiert habe, bestand in der Auswahl der Dinge, die angesichts des begrenzten Platzes nicht in den Abzug aufgenommen werden. Leider ist das Alter nicht immer ein Hinweis darauf, ob etwas dann auch *veraltet* ist. Auch wenn das Meinungsbild auf der Mitgliederversammlung negativ war, wird sich meines Erachtens die Frage nach einer Umstellung auf DVD langfristig immer wieder stellen – spätestens, wenn auch vier CD-ROMs nicht mehr reichen. Ein weiteres Problem, das CTAN an sich betrifft, sehe ich darin, dass CTAN nur dann aktuell sein kann, wenn neue Versionen auch wirklich dort eingereicht werden. Ich finde es unbefriedigend, wenn ich gelegentlich von Paketautoren gesagt bekomme, dass ich lieber die Version von der Homepage des Autors verwenden soll, weil die Version auf CTAN veraltet sei.

Der geneigte Leser hat im vorigen Absatz vielleicht die Wendung „bisheriger Maintainer“ entdeckt. Nachdem ich es letzten Monat bereits vorstandsintern angekündigt habe, folgt hier nun auch die öffentliche Mitteilung, dass ich mich aus zeitlichen Gründen nicht in der Lage sehe, diese Tätigkeit fortzuführen. Dieses Grußwort ist daher auch ein Aufruf an Freiwillige, denn ein Nachfolger ist noch nicht in Sicht. Selbstverständlich biete ich für die Einarbeitung meine Unterstützung an. Die Arbeit am CTAN-Abzug macht übrigens durchaus Spaß und ist auf etwa drei Wochen zeitlich begrenzt, während dieser Zeit jedoch intensiv. Ich hoffe, die CTAN-Abzüge der vergangenen drei Jahre haben Ihnen Freude gemacht.

Mit freundlichem Gruß

Klaus Höppner  
(Stellvertretender Vorsitzender)

## Beschlüsse der 27. Mitgliederversammlung von DANTE e.V. am 5. Oktober 2002 in Augsburg

### Günter Partosch

Zeit:	5. Oktober 2002, ca. 10:15 Uhr–ca. 12:05 Uhr
Ort:	Universität Augsburg Gebäude der Wirtschaftswissenschaftlichen Fakultät Hörsaalzentrum Wirtschaftswissenschaften, Raum 1106 Universitätsstraße 20 86159 Augsburg
Teilnehmer:	33 (anhand der ausgegebenen Stimmkarten)
Leitung:	Volker RW Schaa (Vorsitzender von DANTE e.V.)
Protokollant:	Günter Partosch (Schriftführer von DANTE e.V.)

Die Mitgliederversammlung wurde satzungsgemäß eingeladen und ist beschlussfähig.

### TOP 1: Begrüßung; Vorstellung des Vorstands; Tagesordnung

#### Begrüßung und Tagesordnung

Volker RW Schaa begrüßt die Teilnehmer der 27. Mitgliederversammlung von DANTE e.V. in Augsburg und stellt die Tagesordnung vor:

1. Begrüßung; Vorstellung des Vorstands; Tagesordnung
2. Bericht des Vorstands
3. Förderung von T<sub>E</sub>X-Projekten
  - Bericht über den aktuellen Stand
  - Beschluss über die Bereitstellung neuer Projektmittel
4. Verschiedenes

Sie wird ohne Einspruch akzeptiert.

## Vorstellung des Vorstands

Alle derzeitigen Vorstandsmitglieder sind anwesend und werden von Volker RW Schaa vorgestellt: Volker RW Schaa (Vorsitzender), Klaus Höppner (stellvertretender Vorsitzender), Tobias Sterzl (Schatzmeister), Günter Partosch (Schriftführer) sowie Thomas Koch und Bernd Raichle als Beisitzer.

## TOP 2: Bericht des Vorstands

Tobias Sterzl referiert kurz über die Entwicklung der Mitgliederzahlen:

Anzahl der Mitglieder am Berichtstag	2021
Eintritte in 2002	132
Austritte in 2002	59
(noch) nicht bezahlt haben	101

und die größten Anschaffungen und Ausgaben in diesem Jahr:

DTK, Ausgabe 1/2002 und T <sub>E</sub> Xnische Kontakte	5507,04 €
DTK, Ausgabe 2/2002	2752,11 €
Tagung in Erlangen	1419,00 €
EuroBachoT <sub>E</sub> X (Bursary)	3750,00 €
2 18,3 GB-Festplatten für die Sun-Rechner	494,16 €
1 Büro-Rechner	614,00 €
1 Fotokopierer	972,49 €

Im Übrigen werden unter diesem Tagesordnungspunkt von der MV keine Beschlüsse gefasst.

TOP 3: Förderung von T<sub>E</sub>X-Projekten

## TOP 3.1: Bericht über den aktuellen Stand

Volker RW Schaa berichtet über den aktuellen Stand des *T<sub>E</sub>X Project Funding* und stellt einige Teilprojekte und die Art der Förderung vor:

- L<sub>A</sub>T<sub>E</sub>X
- preview-latex von David Kastrup

- T<sub>E</sub>Xlive-CD-ROM von Sebastian Rahtz und Fabrice Fabineau
- Ausbau und Weiterentwicklung der Font-Familie Antykwa Póławskiego durch die polnische GUST
- pdfExt von Hàn Thê Thành
- CTAN- und T<sub>E</sub>Xlive-CD-ROM

In der folgenden Diskussion kann keine Einigung über den Fortgang der CTAN- und T<sub>E</sub>Xlive-CD-ROM gefunden werden. Ein diesbezüglicher Antrag

*Die MV beauftragt das Präsidium, einen Projekt-Antrag über die Zukunft der CTAN- und T<sub>E</sub>Xlive-CD-ROM zu initiieren.*

wird aber ohne Gegenstimmen und Enthaltungen gebilligt.

### TOP 3.2: Beschluss über die Bereitstellung neuer Projektmittel

Volker RW Schaa begründet die Bereitstellung neuer Projektmittel und stellt den Antrag

*Der Vorstand beantragt die Aufstockung des Projekt-Mittel-Fonds (TBF) auf weitere 10 000 €.*

Er wird ohne Gegenstimmen und Enthaltungen angenommen.

### TOP 4: Verschiedenes

Unter diesem Tagesordnungspunkt berichten u. a.

*Volker RW Schaa* über die *LUG Banking Facility* (LBF), mit dessen Hilfe die einzelnen T<sub>E</sub>X-Gruppen den Zahlungsverkehr untereinander vereinfachen können,

*Klaus Höppner* über den Stand der aktuellen CTAN-CD-ROM – sie ist fertig und wird mit der DTK-Ausgabe 3/2002 verteilt,

*Klaus Höppner* über die nächsten deutschen DANTE-Tagungen (2.–4.4. 2003 in Bremen, 8.–9.9. 2003 in Schloss Rauischholzhausen bei Marburg) und

*Volker RW Schaa* über die nächsten europäischen und weltweiten T<sub>E</sub>X-Tagungen.

Auch hier werden von der Versammlung keine Beschlüsse gefasst.

Volker RW Schaa schließt die Versammlung um ca. 12:05 Uhr.

Volker RW Schaa  
(Versammlungsleiter)

Günter Partosch  
(Protokollant)

# Bretter, die die Welt bedeuten

---

## L<sup>A</sup>T<sub>E</sub>X und WYSIWYG?

preview-latex unter Emacs und andere Ansätze

David Kastrup

*Es schreibt der Griffel und, so er geschrieben,  
fährt fort; nicht Frömmigkeit noch Schläue trieben  
ihn auch nur halbe Zeilen auszutilgen,  
noch könnten Tränen nur ein Wort verschieben.*

OMAR KHAYYÁM

Der anweisungsbasierte Typensatz mit T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X hat seine Leistungsfähigkeit und Flexibilität für eine Vielzahl von Anwendungen wiederholt unter Beweis gestellt. Es wird aber der Mangel an unmittelbarem visuellen Feedback nicht nur bei Anfängern oft als ein Manko bei der Dokumenterstellung und der Korrektur empfunden.

Es existieren mittlerweile eine Reihe von Versuchen, dieses Problem anzugehen, angefangen bei einfachen token-basierten Ansätzen (Syntax Highlighting, Spezialzeichensätze wie mit X-Symbol) bis hin zu kompletten Textverarbeitungssystemen (LyX und T<sub>E</sub>X<sub>MACS</sub>), die L<sup>A</sup>T<sub>E</sub>X im Wesentlichen nur als Exportsprache nutzen.

Eine weitere Klasse von Werkzeugen konzentriert sich nicht auf das Editieren, sondern auf schnellen Zugriff auf die gesetzte Form, meist als seitenorientierte Vorschau in einem separaten Fenster (Whizzy-T<sub>E</sub>X, Instant Preview). Zwar ist die Verarbeitungsgeschwindigkeit dieser Varianten auf heutigen Systemen mehr als ausreichend, dennoch bleibt das Bedürfnis nach einer engeren Kopplung von Quelltext und Vorschau beim Editieren. „Source Specials“ sind ein Werkzeug für eine solche Querverbindung. Das preview-latex-Paket des Autors stellt eine wesentlich engere Kopplung durch die direkte Platzierung von Vorschau-elementen im Quelltext bereit.

## Der Konflikt $\LaTeX$ /WYSIWYG

### Was ist WYSIWYG?

WYSIWYG, eine Kurzform von „what you see is what you get“, ist eigentlich ein Reklameschlagwort, das auf verschiedene Arten von Ähnlichkeit des Eingabefensters mit dem gedruckten Endprodukt angewandt wird.

Streng genommen bedeutet dieser Begriff die Identität von Druckergebnis und Bildschirmanzeige. Dies ist allerdings schlicht unmöglich: Drucker haben von Bildschirmen deutlich unterschiedliche Eigenschaften: andere Auflösungen, andere Kontraste, andere Farb- und Graustufungen. Eine reine Bildschirmkopie eines WYSIWYG-Textverarbeitungssystems wieht, mit einem normalen Ausdruck verglichen, scheußlich aus.

Was wird nun realistischerweise von der WYSIWYG-Bezeichnung erwartet?

*Ähnlichkeit zum Ausdruck:* Das Eingabefenster soll dem Druckbild weitgehend entsprechen.

*Zeichenform:* Zwar ist keine Identität auf der Basis von Pixeln zu erwarten, aber eine Ähnlichkeit in der Form der Einzelzeichen wird generell gefordert. Bildschirmanzeigen haben typischerweise eine weit feinere Intensitätsabstufung als dies beim Druck der Fall sein kann, der dafür aber eine sehr viel höhere Auflösung bietet: das macht Antialiasing sinnvoll. Hierbei wird die niedrige Auflösung dadurch kompensiert, dass der Grauwert von Pixeln entsprechend der Abdeckung mit Tinte, die bei einer feineren Auflösung zustande käme, gewählt wird.

Zeichen mit Antialiasing sind auf dem Bildschirm besser lesbar als dieselben Zeichen ohne Verwendung dieses Tricks, sind aber etwas schwammig und unscharf im Erscheinungsbild, da die mangelnde Auflösung nicht wirklich behoben, sondern nur ihre Auswirkungen auf den Grauwert reduziert wird.

*Erweiterte Zeichensätze:*  $\TeX$  und  $\LaTeX$  selbst nutzen im Wesentlichen einen ASCII-Zeichensatz. Es existieren zwar Erweiterungen, um auch 8-Bit-Zeichensätze ansprechen zu können, hingegen werden etwa mathematische Sonderzeichen wie  $\sum$ ,  $\int$ ,  $\backslash$  als Kontrollsequenzen wie `\sum` eingegeben. Typische WYSIWYG-Systeme böten eine lesbarere Darstellung solcher Zeichen und bequeme Formen der Eingabe, ohne dass die Kenntnis solcher Namen notwendig wäre.

*Nicht-textuelle Elemente:* Am erwähnenswertesten sind hier sicherlich mathematische Formeln und Tafeln. Charakteristisch ist ein Grad an Bedeutungsvermittlung, der mit einer rein textbasierten Darstellung nicht möglich wäre.

*Zeilenumbrüche:* Die meisten WYSIWYG-Systeme haben im Druckergebnis dieselben Zeilenumbrüche wie in der Bildschirmdarstellung.

*Seitenlayout:* Seitenumbrüche und Platzierung von Gleitobjekten werden von WYSIWYG-Systemen ebenfalls häufig entsprechend der Druckdarstellung vorgenommen.

## Der Konflikt

Warum ist das WYSIWYG-Konzept für die meisten dieser Punkte nicht die optimale Lösung, speziell im Zusammenhang mit T<sub>E</sub>X?

*Ähnlichkeit zum Ausdruck:* Da es sich bei T<sub>E</sub>X ja eigentlich um eine Programmiersprache handelt, sind es nicht selten komplexere Konstrukte, die das Endresultat erzeugen. Diese in WYSIWYG-Manier zu bearbeiten ist kaum möglich. WYSIWYG hat die Tendenz, Scheußlichkeiten der Eingabe zu verschleiern: Textverarbeitungssysteme wie Word bieten zwar im Allgemeinen verschiedene Möglichkeiten, die Eingabe zu strukturieren (Formatvorlagen, Textstile usw.), dies ist aber in der Bildschirmdarstellung nicht unmittelbar ersichtlich. Typische Benutzer solcher Systeme neigen dann auch zu eher ungeeigneten Mitteln für die Formatierung ihrer Texte. Dinge wie Leerzeichenorgien für Einrückungen und Anordnungen sieht man nicht gerade selten; solches Gebastel ist gegenüber Textänderungen und Wechsel von verwendeten Druckern und/oder Fonts sehr empfindlich. T<sub>E</sub>X bietet einen einfachen Weg, Kommentierungen in den Quelltext einzufügen (% leitet Kommentarzeilen ein). Solche Kommentare sind in WYSIWYG-Systemen nicht einsetzbar. Das führt aber auch dazu, dass man nicht mehr nachvollziehen kann, was man selbst oder der Autor sich gedacht hat: auch das Lernen aus Vorlagen geübterer Benutzer ist stark erschwert.

*Zeichenform:* Die meistgenutzte Zeichensatzfamilie im Umfeld von T<sub>E</sub>X ist die Computer Modern family von Knuth. Für diese Fonts ist eine feine Balance zwischen verschiedenen Strichbreiten und Haarlinien charakteristisch, die zum einen zu einer „geschlossenen“ Grundform der Zeichen führt, zum anderen aber im Gesamtgrauwert die „führende“ Eigenschaft

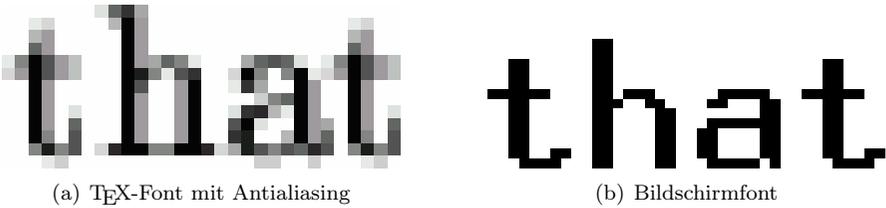


Abbildung 1: Fonts in Bildschirmauflösung

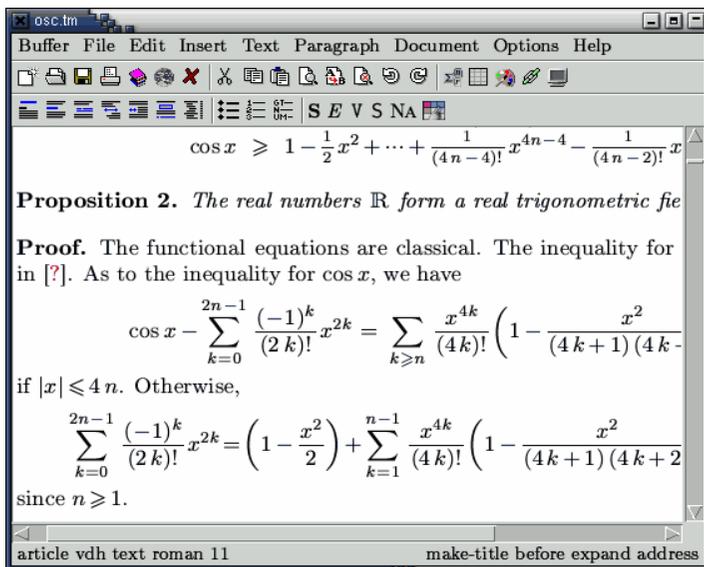
ihrer Serifen nicht verliert (wie aus den „t“s im Artikel selbst hervorgehen sollte). Das „t“ hat aus der Nähe betrachtet eine geschlossene Kurve, aber die Strichbreitenverteilung lässt im Gesamtgraubild den Einfluss der schließenden Haarlinie schwinden (Abbildung 1(a)).

Solche Feinheiten sind bei verfügbaren Bildschirm-Auflösungen nicht darstellbar. Im Interesse der Lesbarkeit sind deswegen für die Bildschirmdarstellung konzipierte Fonts vorzuziehen.

*Erweiterte Zeichensätze:* Die Verbesserung der Lesbarkeit ist offensichtlich, die Umsetzung bereitet keine prinzipiellen Probleme.

*Nicht-textuelle Elemente:* Diese profitieren von einer WYSIWYG-artigen Darstellung am meisten.  $\TeX$  bietet für den Formelsatz mächtige Möglichkeiten, und eine Vielzahl von Zusatzpaketen nutzen seine Programmierungsmöglichkeiten, um zusätzliche Funktionalität bereitzustellen. Jedoch erfordern Arrangements wie Formeln und Tabellen erheblichen Programmieraufwand für eine adäquate Eingabedarstellung im Editor. Eine direkte Verfügbarkeit einer solchen Darstellung erleichtert es erheblich, einen Gedankengang zu entwickeln und ist auch für Korrekturarbeiten sehr wünschenswert.

*Zeilenumbrüche und Seitenlayout:*  $\TeX$  nutzt ein absatzumfassendes Optimierungsverfahren, um seine Zeilenumbrüche festzulegen und ein lokales Optimierungsverfahren zur Festlegung der Seitenumbrüche. „Insertions“ dienen zur Platzierung zusätzlichen Seitenmaterials wie Fußnoten und Abbildungen. Diese Prozesse werden durch eine umfangreiche Liste von Strafpunkten gesteuert, die für verschiedene auch über mehrere Zeilen reichende Merkmale festgelegt werden (optische Kompatibilität des Rand-

Abbildung 2: Typisches T<sub>E</sub>X<sub>MACS</sub>-Fenster

ausgleiches aufeinander folgender Zeilen, Strafen für gehäuft auftretende Trennungen usw).

Die nichtlokalen Optimierungen von T<sub>E</sub>X machen sehr gute Resultate möglich. Diese ständig bei der Texteingabe durchzuführen war früher aus Geschwindigkeitsgründen nicht praktikabel. Auch wo dies möglich wäre, sind jedoch die potenziell großen Anzeigeänderungen bei kleinen Eingabeänderungen für den Eingabefluss eher störend.

## Die Überwindung der Kluft

### T<sub>E</sub>X<sub>MACS</sub>

Das Textverarbeitungssystem T<sub>E</sub>X<sub>MACS</sub> hinterlässt gegenwärtig einen guten Eindruck, wie eine umfassende Umsetzung von WYSIWYG-Konzepten für T<sub>E</sub>X aussehen könnte. Zwar nutzt T<sub>E</sub>X<sub>MACS</sub> weder T<sub>E</sub>X noch bietet es einen unmittelbaren Zugang zu T<sub>E</sub>X-Paketen und -Programmierung, verwendet aber Textsatzalgorithmen von T<sub>E</sub>X und bietet T<sub>E</sub>X-Fonts mit Antialiasing

auf dem Bildschirm. Das Druckbild stimmt mit der Bildschirmdarstellung weitestgehend überein, wenngleich der Seitenumbruch im Quelltext optional ist. T<sub>E</sub>X<sub>MACS</sub> unterstützt zwar L<sup>A</sup>T<sub>E</sub>X nicht unmittelbar als Dateiformat, kann aber L<sup>A</sup>T<sub>E</sub>X exportieren und, deutlich eingeschränkter, auch importieren. Die Tastaturbelegung erinnert an Emacs, und einige T<sub>E</sub>X-ähnliche Kontrollsequenzen können mit dem Backslash eingeleitet werden. Der Editor kann in Guile, dem vom GNU-Projekt eingesetzten Dialekt der Lisp-verbundenen Sprache Scheme, erweitert und angepasst werden. Für den Export nach L<sup>A</sup>T<sub>E</sub>X steht auch die Möglichkeit der Einbindung von L<sup>A</sup>T<sub>E</sub>X-Code, der ohne Interpretation exportiert wird, zur Verfügung.

Auf einem 200 MHz-System wirkte T<sub>E</sub>X<sub>MACS</sub> etwas träge. Während die meisten L<sup>A</sup>T<sub>E</sub>X-Nutzer nach einiger Eingewöhnung mit T<sub>E</sub>X<sub>MACS</sub> wohl halbwegs klarkommen werden, ist eine intensivere Nutzung der vielfältigen Möglichkeiten von L<sup>A</sup>T<sub>E</sub>X eher unhandlich, und der Austausch von Dokumenten und die Zusammenarbeit mit anderen L<sup>A</sup>T<sub>E</sub>X-Nutzern ist sicher problematisch.

Persönlich empfinde ich den „Ziehharmonika-Effekt“ störend: Der bei der Texteingabe kontinuierlich erfolgende Randausgleich und Neuumbuch führt zum wiederholten Schrumpfen und Wachsen der Wortabstände bei der Textbearbeitung. Das unmittelbare Feedback von Tastendruck zu endgültiger Anzeige ist sicherlich bei letzten Feinarbeiten am typographischen Ergebnis nützlich, der Enthusiasmus, mit dem hier ganze Absätze am laufenden Band umgeschichtet und ausgerichtet werden während man einzelne Buchstaben tippt, erscheint so sinnvoll und praktisch wie ein unermüdlicher Trupp Fensterputzer auf einer Baustelle.

## L<sub>y</sub>X

Ähnlich wie bei T<sub>E</sub>X<sub>MACS</sub> handelt es sich bei L<sub>y</sub>X um ein komplettes Textverarbeitungssystem. Im Unterschied zu T<sub>E</sub>X<sub>MACS</sub> wird hier allerdings das Schlagwort WYSIWYM (what you see is what you mean) statt WYSIWYG propagiert. Während in früheren Versionen dieser Begriff vornehmlich als Euphemismus für „was man sieht, ist irgendwie ähnlich zu dem, was rauskommt“ herhielt, wurde gerade in neueren Versionen eine Menge getan, um in der Anzeige tatsächlich vielfältige Information über die Dokumentstruktur zu vermitteln. Da L<sub>y</sub>X nicht daran gebunden ist, die Ausgabe entsprechend der Eingabe zu gestalten, kann es durch Verwendung von Farben und grafischen Elementen für Dinge wie Querverweise vielseitiger und flexibler Information vermitteln, als das bei reinem WYSIWYG der Fall wäre.

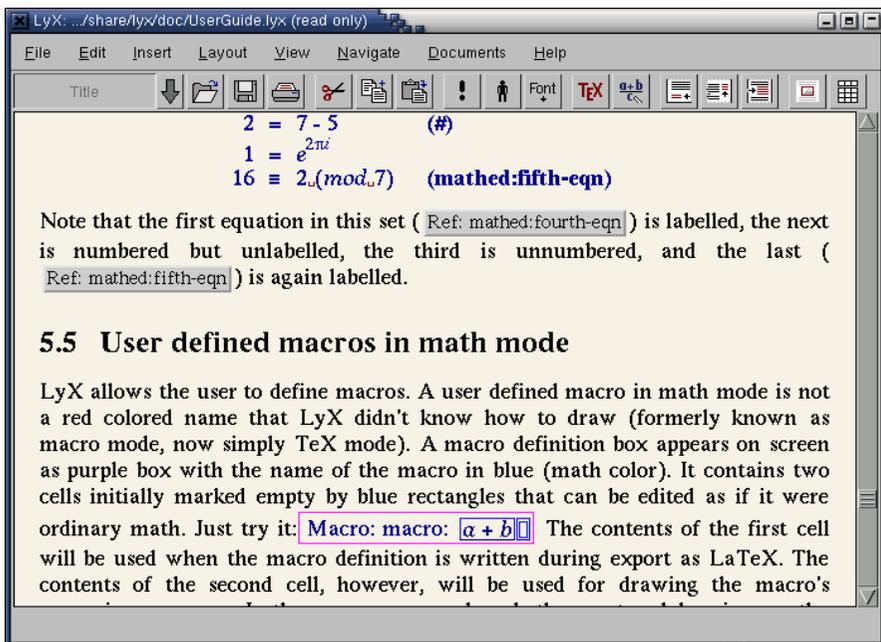


Abbildung 3: Typisches LyX-Fenster

Zeichensatzumschaltungen im Dokument finden zwar ihren Niederschlag in der Bildschirmdarstellung, diese greift aber auf Fonts zurück, die sich von den beim Druck verwendeten unterscheiden. Die Bildschirmfonts sind als Teil der Benutzereinstellungen konfigurierbar, wozu Kenntnisse des Fontsystems von X11 anzuraten sind. Die Standardeinstellungen scheinen aber gerade im Mathematikbereich nicht auf Lesbarkeit hin optimiert.

LyX nutzt L<sup>A</sup>T<sub>E</sub>X für den Textsatz beim Druck, wobei es auf Standardklassen und -pakete zurückgreift. Der L<sup>A</sup>T<sub>E</sub>X-Export gestaltet sich damit unproblematisch; der Import hingegen zeitigt gemischte Resultate. Im Allgemeinen ist nicht einmal der unveränderte Reimport von Dokumenten, die LyX selbst exportiert hat, möglich. LyX hat ebenfalls ein hauseigenes Dateiformat. Eine gemeinsame Erstellung von L<sup>A</sup>T<sub>E</sub>X-Dokumenten mit anderen Autoren ist damit auch hier kaum praktikabel. L<sup>A</sup>T<sub>E</sub>X-Konstrukte, die LyX nicht bekannt sind, können direkt in das Dokument eingefügt werden und werden dann als

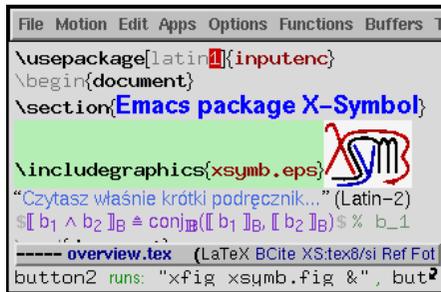


Abbildung 4: X-Symbol in vollem Schwung

„ERT“ („Evil Red Text“) angezeigt und unverändert exportiert. Es ist ein erklärtes Ziel der LyX-Entwickler, nach und nach ERT weitgehend überflüssig zu machen, indem LyX mehr und mehr Konstrukte von Haus aus versteht. Eine der Stärken von LyX ist der Mathematikeditor, der dem von TeX<sub>MACS</sub> ähnelt und einen recht großen Umfang von LaTeX- und AMS<sub>LaTeX</sub>-Konstrukten von Haus aus unterstützt.

LyX hat ebenfalls eine tastendruckorientierte Bildschirm-Update-Strategie, die in Verbindung mit Randausgleich auf dem Bildschirm zum oben beschriebenen Ziehharmonika-Effekt bei Textergänzungen führt.

## X-Symbol

X-Symbol ist das erste hier vorgestellte System, das nicht eine komplette Textverarbeitung implementiert, „nur“ um seine WYSIWYG-Konzepte für den TeX-basierten Satz umzusetzen. Es ist wohl kein Zufall, dass es, wie die anderen, ohne eigens implementierten Editor daherkommenden, vorgestellten Lösungen, unter dem erweiterbaren Editor Emacs (oder der Variante XEmacs) läuft.

Emacs ist

- für das Editieren von Quelltexten vorgesehen
- Freie Software
- eine mächtige Editierumgebung

- einfach mit einer sehr interaktiven Programmiersprache (Emacs-Lisp) erweiterbar
- für eine Vielzahl von Betriebssystemen erhältlich.

Da T<sub>E</sub>X ein auf verschiedensten Systemen vorhandenes, freies Software-System mit editierbarer Quelltextsprache ist, passt es mit Emacs recht gut zusammen. Das meistverwendete Emacs-Paket AUCT<sub>E</sub>X stellt eine umfangreiche Editier- und Laufzeitumgebung für T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X bereit.

Was ist jetzt der Einsatzbereich von X-Symbol? Es kümmert sich weitgehend um den WYSIWYG-Aspekt „erweiterter Zeichensatz“. Quelltexte wie `\sum` und `\leq` werden als „ $\sum$ “ und „ $\leq$ “ dargestellt. Diakritische und sonstige Sonderzeichen des Textmodus wie `"a` werden in lesbarer Form wie „ä“ dargestellt. Hierbei beschränkt sich X-Symbol auf den Bereich einfacher, kurzer Kontrollsequenzen. Einige der dabei eingesetzten Zeichensätze sind Bestandteil des X-Symbol-Paketes selbst; die verwendeten Fonts fügen sich in die normalen Nichtproportionalschriften von Emacs gut ein. Das bedeutet u. a., dass Operatoren wie  $\int$  auf dem Bildschirm recht klein daherkommen. Eine zusätzliche Option ist die Verwendung spezieller Fonts, um im Zusammenspiel mit der farblichen Syntaxhervorhebung von Emacs hoch- und tiefgestellte Terme mit kleineren Zeichen und entsprechender Positionierung darzustellen.

X-Symbol führt diese Konvertierungen unmittelbar bei der Klartexteingabe dieser Kontrollsequenzen durch, kann aber diese Sonderzeichen auch über eine Vielzahl anderer Eingabemethoden (etwa menubasiert) einfügen. Die Zeichen im Quelltext werden hierbei komplett ersetzt und lediglich beim Schreiben der Textdateien wieder transparent in ihre T<sub>E</sub>X-Äquivalente konvertiert. Dies führt zu ein paar neuralgischen Punkten, an denen Fehlfunktionen permanente Dateiänderungen verursachen können.

- Falls durch einen Programmfehler der Text beim Schreiben nicht in seine T<sub>E</sub>X-geeignete Form überführt wird, bekommt man relativ unlesbaren Datensalat. Die Daten sind zwar noch vorhanden, ihre Rekonstruktion erfordert aber einigen Aufwand.
- Die Kombination der beim Einlesen und Wegschreiben der Quelldatei durchgeführten Konvertierungen führt nicht notwendigerweise zum selben Ergebnis. Dies ist in verbatim-ähnlichen Umgebungen besonders problematisch. Die neuesten Versionen von X-Symbol gehen hier aber wesentlich vorsichtiger und vorhersehbarer als frühere Versionen vor.

Da die interne Darstellung des Textes bei diesen Konvertierungen geändert wird, können Editoroperationen wie Suchen und Ersetzen erschwert oder unpraktikabel werden.

Ein Bestandteil von X-Symbol ist die Bereitstellung eines Ersatzmechanismus für L<sup>A</sup>T<sub>E</sub>Xs Paket `inputenc` auf Editorebene (was für plainT<sub>E</sub>X-Nutzer noch interessanter sein könnte). Das ist eine interessante Option gerade bei solchen Texten, die sonst eine wiederholte Umschaltung der Eingabekodierung im Dokument erfordern könnten, was eine konsistente Textdarstellung im Editor recht schwierig gestaltet. Da mit X-Symbol alle Zeichen der verschiedenen Kodierungen in ASCII-Kontrollsequenzen beim Speichern und Laden umgesetzt werden, wird dieses Problem damit umgangen.

Der größte Nutzwert von X-Symbol liegt vermutlich bei der extensiven Eingabeunterstützung, die von Tastaturabkürzungen, über das „Grid“ (ein großes Menü mit allen verfügbaren Zeichen in einem separaten Editorfenster) und Eingabe bis zur Menüleiste von Emacs reicht. Abgesehen vom Grid sind diese Methoden nicht wirklich mit WYSIWYG verwandt.

X-Symbol ist ein Werkzeug, das sich mit der Eingabe beschäftigt, mit ihrer Darstellung, Umwandlung und Beschleunigung: Previewfunktionalität im eigentlichen Sinne ist nicht enthalten. Speziell nicht behandelt werden komplexere Anordnungen von Zeichen, wie mathematische Formeln sie darstellen. Damit ist der abgedeckte Umfang erheblich geringer als bei Systemen wie Ly<sub>X</sub>. Auf der anderen Seite stellt es einem aber beim Durchgriff auf die volle Leistungsfähigkeit von L<sup>A</sup>T<sub>E</sub>X keine Hindernisse oder Unbequemlichkeiten in den Weg. Die Umsetzung der Symbole kann es nicht nur für L<sup>A</sup>T<sub>E</sub>X, sondern auch für plainT<sub>E</sub>X und SGML oder HTML durchführen.

Abbildung 5(b) zeigt ein typischeres Beispiel für die Bildschirmdarstellung von X-Symbol, hier im Zusammenspiel mit *preview-latex*.

### *preview-latex*

Auch bei *preview-latex* handelt es sich um eine Emacs-Erweiterung. Im Gegensatz zu X-Symbol ist sein Einsatzbereich ausschließlich auf den Anzeigebereich von WYSIWYG beschränkt. Es ändert den eigentlichen Quelltext in den Textspeichern von Emacs nicht und hat keinerlei Einfluss auf die externe Dateidarstellung. Während X-Symbol sich mit dem Ersatz einzelner Kontrollsequenzen mit Buchstaben und Symbolen, die zum Bildschirmfont

Normalverteilung sich unabhängig  
wir deswegen das Resultat

```
\begin{equation}
\label{eq:4}
f(x,t)=\int_{-\infty}^{\infty}f(x',0)\sqrt{2\pi}\exp\biggl(-\frac{\bigl(x-\mu_f(\mu_0=x',t)\bigr)^2}{2\sigma_f^2}\biggr)\,dx'
\end{equation}
```

Mit der Definition des \rlq Sch

(a) Gleichung offen

Normalverteilung sich unabhängig  
wir deswegen das Resultat

```
\begin{equation}
\label{eq:4}
f(x,t)=\int_{-\infty}^{\infty}\frac{f(x',0)}{\sqrt{2\pi\sigma_f^2(\sigma_0^2=0,t)}}\exp\biggl(-\frac{\bigl(x-\mu_f(\mu_0=x',t)\bigr)^2}{2\sigma_f^2}\biggr)\,dx'
\end{equation}
```

Mit der Definition des \rlq Sch

(b) Mit X-Symbol

Normalverteilung sich unabhängig  
wir deswegen das Resultat

$$f(x,t) = \int_{-\infty}^{\infty} \frac{f(x',0)}{\sqrt{2\pi\sigma_f^2(\sigma_0^2=0,t)}}$$

Mit der Definition des \rlq Sch

(c) Geschlossen

Abbildung 5: Verwendung von *preview-latex* mit und ohne X-Symbol

passen, befasst, ersetzt *preview-latex* die Darstellung komplexer Textelemente (Formeln, Abschnittsüberschriften, Abbildungen, eingefügte Grafiken) mit einer akkuraten und mit Antialiasing versehenen Druckvorschau, die durch Übersetzung der Textvorlage unter Einsatz von L<sup>A</sup>T<sub>E</sub>X, *dvips* und *GhostScript* erstellt wird. Ein einziger Aufruf von L<sup>A</sup>T<sub>E</sub>X (potenziell mit einem vorher erzeugten speziellen Format) erzeugt die Druckvorschauen für das komplette Dokument, und ein einziger Aufruf von *GhostScript*, das als Daemon betrieben wird, setzt die von *dvips* produzierte PostScript-Datei mit den Druckvorschauen in Grafikdateien um, wobei die augenblicklich auf dem angezeigten Bildschirmbereich darzustellenden Vorschauen vorgezogen werden. Zukünftige Versionen werden durch Zusatzprogramme *dvips* und *GhostScript* weitgehend umgehen; aber schon jetzt ist die Bearbeitungsgeschwindigkeit auch auf eher bescheidener Hardware recht erträglich. Eine weitere geplante Option ist die Möglichkeit, L<sup>A</sup>T<sub>E</sub>X und *dvips* durch *pdfL<sup>A</sup>T<sub>E</sub>X* zu ersetzen.

Die Vorschauen werden in Farbe und Größe der Standarddarstellung im Emacs-Fenster angepasst, sind aber ansonsten identisch zur gedruckten Version. Da *preview-latex* keine Kenntnis von der inneren Struktur solcher Textelemente hat, wird beim Editieren auf die normale Textdarstellung umgeschaltet und auf die grafische Vorschau erst wieder zurückgeschaltet, wenn der Anwender die Beendigung seiner Editierarbeit signalisiert.

Welche Textelemente einer Vorschau unterworfen werden, wird durch ein externes L<sup>A</sup>T<sub>E</sub>X-Paket entschieden. Dessen Arbeitsweise kann leicht durch Deklarationen im Dokumentvorspann oder in einer separaten Konfigurationsdatei vorgegeben werden.

$$d(x) = \frac{1}{\sqrt{2\pi}\sigma_d(t)} \exp\left(-\frac{(x-\mu_d(t))^2}{2\sigma_d^2(t)}\right)$$

(a) TeXMACS

$$d(x) = \frac{1}{\sqrt{2\pi}\sigma_d(t)} \exp\left(-\frac{(x-\mu_d(t))^2}{2\sigma_d^2(t)}\right)$$

(b) LyX

```
\documentclass[12pt, a4paper]{article}
\begin{document}

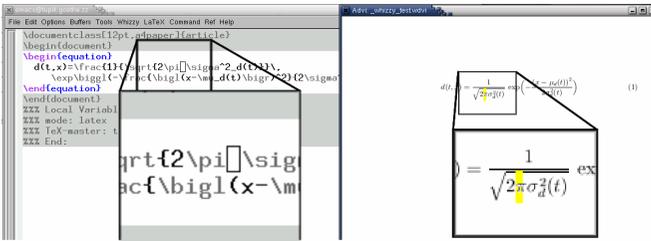
$$d(t, x) = \frac{1}{\sqrt{2\pi}\sigma_d^2(t)} \exp\left(-\frac{(x-\mu_d(t))^2}{2\sigma_d^2(t)}\right)$$

\end{document}
```

(c) preview-latex

```
\begin{equation}
d(t,x)=\frac{1}{\sqrt{2\pi}\sigma_d(t)}\exp\left(-\frac{(x-\mu_d(t))^2}{2\sigma_d^2(t)}\right)
\end{equation}
```

(d) X-Symbol



(e) Whizzy-TeX

Abbildung 6: Detailsichten

Die Arbeitsprinzipien von preview-latex sind nicht spezifisch für Emacs: ähnliche Funktionalität wird gegenwärtig für LyX implementiert, momentan auf den Mathematikmodus und die `\input`-Anweisung beschränkt.

### Whizzy-TeX

Whizzy-TeX ist eines von mehreren Systemen, die die automatische schnelle Neugenerierung von Druckvorschauen in einem separaten Arbeitsfenster zum Ziel haben. Es ist (welche Überraschung) als Emacs-Paket implementiert und wird am besten in Zusammenarbeit mit dem Previewer Active-DVI (in Objective CAML geschrieben) vom selben Autor eingesetzt, der die korrekte Seitenposition ohne Flackern ansteuern und auffrischen kann, wenn sich die

DVI-Datei ändert. Zwar ist Whizzy- $\text{T}_{\text{E}}\text{X}$  auch mit  $\text{x}\text{dvi}$  einsetzbar; die Bildschirmdarstellung ist dann aber deutlich unruhiger.

Whizzy- $\text{T}_{\text{E}}\text{X}$  verfolgt kontinuierlich die augenblickliche Textposition und Textänderungen und lässt im Falle von Änderungen  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  von einem nahen Punkt, an dem es ein Format bei einem vorangehenden Lauf erzeugt hat, neu laufen. Auf diese Weise erfolgt die Nachführung der Druckvorschau recht zügig, und man kann interaktiv mit Positionierungen und anderen Layout-Entscheidungen experimentieren.

In Abbildung 6(e) kann man die Zusammenarbeit von Whizzy- $\text{T}_{\text{E}}\text{X}$  mit Active-DVI sehen.

### Active $\text{T}_{\text{E}}\text{X}$ /Instant Preview

Das Arbeitsprinzip von Active $\text{T}_{\text{E}}\text{X}$  besteht in einem kontinuierlich laufenden  $\text{T}_{\text{E}}\text{X}$ -Prozess, dem  $\text{T}_{\text{E}}\text{X}$ -Daemon, der auf Anfrage Seiten setzt. Ein separates Programm extrahiert dann einzelne Seiten aus der dabei laufend wachsenden DVI-Datei.

Das Instant-Preview-Paket für Emacs nutzt dies für den Update eines Vorschaufensters auf der Zeitbasis einzelner Tastendrucke. Für dieses Arbeitsprinzip sind im Wesentlichen zustandslose Quelldokumente Voraussetzung, damit der wiederholte Durchlauf einzelner Passagen keine Probleme bereitet. Aus diesem Grund ist dieses System für die Verwendung mit  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  weitgehend ungeeignet. Der Autor des Systems nutzt für die Bildschirmdarstellung  $\text{x}\text{dvi}$ ; eine Verwendung von Whizzy- $\text{T}_{\text{E}}\text{X}$ s Previewer Active-DVI würde vermutlich eine ruhigere Bildschirmdarstellung zur Folge haben.

Hauptanwendungsziel von Active $\text{T}_{\text{E}}\text{X}$  ist hohe Verarbeitungsgeschwindigkeit auch auf bescheidener Hardware. Es bietet darüber hinaus wenig Vorteile gegenüber Whizzy- $\text{T}_{\text{E}}\text{X}$  und verfolgt ähnliche Ziele, wobei es für  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  wesentlich ungeeigneter ist.

### Source Specials

„Source Specials“ sind ein Verfahren zur Kopplung von Editor und Previewer, das durch Platzierung spezieller Markierungen in der DVI-Datei die zugehörige Quelltextposition angibt. Diese Marken werden entweder durch einen speziellen  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Stil oder automatisch bei Verwendung neuerer  $\text{T}_{\text{E}}\text{X}$ -Versionen durch Setzen einer Kommandozeilenoption erzeugt. Ihr einziger Zweck ist

die Quernavigation. In Verbindung mit Unterstützung in Editor und Previewer kann man Vorwärtssuche (die Position im Editorfenster wird vom Vorschauenfenster automatisch mitverfolgt) und Rückwärtssuche (ein Klick im Vorschauenfenster positioniert den Editor auf die zugehörige Quelltextstelle) implementieren. Von diesem reinen Navigationswerkzeug wird eigentlich keine WYSIWYG-Funktionalität bereitgestellt; es zeigt aber den Bedarf einer engeren Kopplung von Quelltext und Vorschau.

## Resümee

Abbildung 6 zeigt Bildschirmaufnahmen der verschiedenen Behandlung einer Formel durch die gegebenen Systeme (*preview-latex* zeigt hierbei nur die Formel im geschlossenen Zustand, siehe deswegen auch Abbildung 5). Eine Zusammenfassung ist in Tabelle 1 zu finden.

Diejenigen WYSIWYG-Systeme, die eine gebräuchlichere Editierumgebung zur Verfügung zu stellen versuchen, haben mit einigen Schwierigkeiten zu kämpfen:

- Sie sind darauf angewiesen, eine komplette Editierumgebung selbst zur Verfügung zu stellen.
- Sie müssen zumindest die Bildschirm-Interpretation aller von ihnen unterstützten Konstrukte selbst beherrschen und können dies nicht an L<sup>A</sup>T<sub>E</sub>X delegieren. Auf diese Weise kann aber immer nur eine Untermenge von L<sup>A</sup>T<sub>E</sub>X korrekt und effizient unterstützt werden. Beide hier vorgestellten Produkte verwenden ihre eigenen Dateiformate, wodurch L<sup>A</sup>T<sub>E</sub>X als Format zum Dokumentaustausch weitgehend ungeeignet wird. In Abbildung 6 kann man den Umgang dieser Systeme mit unbekanntem Größen-deklarationen im Mathematikmodus sehen.
- Eine ständig aktuelle und konsistente Eingabedarstellung ist nötig, was den tatsächlichen Einsatz von L<sup>A</sup>T<sub>E</sub>X zur Generierung der Anzeige weitgehend unpraktikabel macht.

Systeme mit tastendruckweiser Umformatierung samt Randausgleich im Eingabefenster erscheinen mir für die kontinuierliche Texteingabe irritierend. Weiterhin kostet ein solches Vorgehen auf langsameren Systemen nicht unerhebliche Rechenleistung, und führt zu Unruhe im Eingabefenster. Die Auswirkungen auf größere Flächen sind bei der Texterstellung selbst eher lästig, bei der Endkorrektur von Dokumenten aber sinnvoll. Zwar führt

Whizzy- $\text{T}_{\text{E}}\text{X}$  gleichfalls eine tastendruckorientierte, kontinuierliche Neuformatierung durch; durch deren Anordnung in einem separaten Fenster ist diese aber mit erheblich weniger Ablenkung verbunden. Allerdings ist die von dieser Kombination beanspruchte Bildschirmfläche wesentlich größer und erfordert zur Bearbeitung einen Blickwechsel. `preview-latex` geht mit Bildschirm-Auffrischungen (die nur auf Anweisung erfolgen), Bildschirmfläche und Aufmerksamkeitslenkung sehr ökonomisch um, wobei es allerdings keine Vorschau eines Objektes während der Änderungen bietet. Eine gewisse Verbesserung kann man deswegen durch zusätzliche Nutzung von X-Symbol bei der Eingabe erreichen.

Die Kombination von  $\text{L}_{\text{Y}}\text{X}$ ' Mathematikeditor mit der kommenden `preview-latex`-verwandten Vorschauunktionalität ist sicherlich eine interessante Entwicklung. Erfahrene  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Nutzer mögen die direkte Eingabe der Befehle vorziehen; die Kombination eines leichten Zugriffes sowohl auf eine bei der Eingabe gut arrangierte Darstellungsform, als auch einer exakten, von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  selbst generierten Vorschau als Alternative ist sicher für fortgeschrittene Nutzer bei der Erforschung über den von  $\text{L}_{\text{Y}}\text{X}$  direkt unterstützten Funktionsumfang hinaus interessant.

Auf der Eingabeseite von WYSIWYG erhält man mit X-Symbol einen guten Kompromiss. Auf der Vorschauseite erscheint – verlässt man das Feld der spezialisierten Textverarbeitungssysteme, bei denen  $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$  die vielleicht vollständigste Umsetzung bietet – das Arbeitsprinzip von `preview-latex` als für die Implementation in andere Editiersysteme am geeignetsten: Die Verwendung von  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  für den exakten Textsatz garantiert hohe Detailtreue, ohne dass die volle Kontrolle des Benutzers über die eingesetzten  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Mittel eingeschränkt wird. Für die Art der von `preview-latex` unterstützten syntaktischen Einheiten ist der Verzicht auf eine tastendruckorientierte Neuanzeige (unter den beschriebenen Werkzeugen eher ungewöhnlich) eher von Vorteil, da damit das Textelement ohne Ablenkung zusammengestellt werden kann und vom Benutzer erst dann  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  überantwortet wird, wenn er der Meinung ist, dass dieses syntaktisch korrekt ist. Für interaktive Korrekturen im Zusammenhang mit dem Endabgleich des Satzspiegels ist die schnelle Nachführung einer separaten seitenorientierten Darstellung, wie sie Whizzy- $\text{T}_{\text{E}}\text{X}$  vornimmt, sicher sinnvoll. Der zusätzliche Verbrauch an Bildschirmfläche und der Aufmerksamkeitswechsel zwischen Quelltext und Vorschau machen es für umfangreichere Redigierarbeiten und die Texterstellung selbst weniger geeignet. Die Teilung der Aufmerksamkeit wird in Abbildung 6(e) offensichtlich: Keine andere Bildschirmaufnahme erforderte zur Illustration des Arbeitsprin-

zips den Einsatz von lokalen Vergrößerungen. Ein Vorteil von Whizzy- $\text{T}_{\text{E}}\text{X}$  ist sicherlich, dass seine Implementation der Druckvorschau recht leicht auf andere Systeme übertragbar sein sollte: eine Einbindung etwa in  $\text{L}_{\text{Y}}\text{X}$  erscheint praktikabel.

## Zukünftige Entwicklungen

Die vorgestellten vollständigen Textverarbeitungssysteme werden sicherlich mehr und mehr Konstrukte von  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  unterstützen. Weitere Geschwindigkeitsverbesserungen werden wohl keine wesentlichen Auswirkungen auf die Arbeit mit den vorgestellten Systemen haben. Die interessantesten Ansätze sind sicherlich die Mischformen, die vermutlich in Zukunft zunehmend auch außerhalb von Emacs zu finden sein werden. Zum Beispiel wird  $\text{L}_{\text{Y}}\text{X}$  in seiner nächsten Version (vermutlich 1.3.0) preview-latex-ähnliche Funktionalität aufweisen, zunächst im Wesentlichen auf den Mathematikmodus beschränkt.

Zukünftige separat verwendbare  $\text{T}_{\text{E}}\text{X}$ -Komponenten (Omega-Bibliotheken, DVI-konvertierende Kontrollprozesse) könnten Einzug in andere Textprozessoren halten. Speziell für den Mathematikmodus wäre eine solche Einbettung von  $\text{T}_{\text{E}}\text{X}$ -Fonts und Algorithmen denkbar.

## Verfügbarkeit

Hier gibt es die vorgestellten Systeme:

$\text{T}_{\text{E}}\text{X}_{\text{MACS}}$	<a href="http://www.texmacs.org">http://www.texmacs.org</a>
$\text{L}_{\text{Y}}\text{X}$	<a href="http://www.lyx.org">http://www.lyx.org</a>
X-Symbol	<a href="http://x-symbol.sourceforge.net">http://x-symbol.sourceforge.net</a>
preview-latex	<a href="http://preview-latex.sourceforge.net">http://preview-latex.sourceforge.net</a>
Whizzy- $\text{T}_{\text{E}}\text{X}$	<a href="http://pauillac.inria.fr/whizzytex">http://pauillac.inria.fr/whizzytex</a>
Active-DVI	<a href="http://pauillac.inria.fr/advi">http://pauillac.inria.fr/advi</a>
Active $\text{T}_{\text{E}}\text{X}$	<a href="http://www.activetex.org">http://www.activetex.org</a>
Emacs	<a href="http://www.gnu.org/software/emacs">http://www.gnu.org/software/emacs</a>
XEmacs	<a href="http://www.xemacs.org">http://www.xemacs.org</a>
Source Specials	<a href="http://xdvi.sourceforge.net/inverse-search.html">http://xdvi.sourceforge.net/inverse-search.html</a>
Omega	<a href="http://omega.cse.unsw.edu.au">http://omega.cse.unsw.edu.au</a>

Alle vorgestellten Systeme stehen unter der GNU General Public License und sind deswegen Freie Software ( $\text{L}_{\text{Y}}\text{X}$  ist unter einer modifizierten Variante lizenziert, um das Linken mit der XForms-Bibliothek zu gestatten).

Tabelle 1: Bewertung der Systeme

	T <sub>E</sub> X <sub>MACS</sub>	L <sub>Y</sub> X	X-Symbol	preview- latex	Whizzy- T <sub>E</sub> X
Portierbarkeit	-	-	-	0	+
Quelltexterhalt	-	-	+	++	++
Kontinuierliche Arbeitsweise	ja	ja	ja	nein	ja
Arbeit im Quelltextfenster	ja	ja	ja	ja	nein
Editiergeschwindigkeit	-	0	+	N/A	N/A
Editierbildtreue	++	0	-	+	N/A
Vorschau- geschwindigkeit	++	0	N/A	+	+
Editierflächenbedarf	+	+	+	+	-
Vorschauflächenbedarf	+	-	N/A	+	-
Quernavigation	++	-	-	+	0

Eingesetzte Kategorien:

*Portierbarkeit* Wie leicht lässt sich das Arbeitsprinzip auf andere Editoren und Editierumgebungen übertragen?

*Quelltexterhalt* Wie zuverlässig bleibt existierender Quelltext beim Editieren erhalten?

*Kontinuierliche Arbeitsweise* Arbeitet das System durchgängig während der Texteingabe?

*Arbeit im Quelltextfenster* Arbeitet das System direkt im Quelltextfenster?

*Editiergeschwindigkeit* Wie schnell reagiert das System auf Tastendrücke bei der normalen Texteingabe? N/A falls das System dabei nicht beteiligt ist.

*Editierbildtreue* Wie nah am gesetzten Ergebnis ist die Darstellung im Quelltextfenster?

*Vorschaugeschwindigkeit* Wie schnell sind echte Vorschauen verfügbar?

*Editierflächenbedarf* Welchen Bildschirmflächenverbrauch hat das System beim Editieren?

*Vorschauflächenbedarf* Welchen Bildschirmflächenverbrauch hat das System, wenn eine echte Vorschau benötigt wird? N/A für Systeme ohne eigene Vorschaumethode.

*Quernavigation* Wie zugänglich ist die Verbindung von echter Vorschau und Quelltext?

Beschreibung der Bewertungen:

- ++ Sehr gut
- + Gut
- 0 Zufriedenstellend
- Lässt zu wünschen übrig
- N/A Nicht anwendbar

## Tipps und Tricks: Immer im Rahmen bleiben

Rolf Niepraschk

Zum Hervorheben bestimmter Textbereiche macht man gelegentlich gern von Einrahmung oder Einfärben des Hintergrunds Gebrauch. Üblicherweise setzt man dazu in einem L<sup>A</sup>T<sub>E</sub>X-Dokument den betreffenden Text in eine minipage, welche einer fbox- oder colorbox-Anweisung übergeben wird. Hier ein Beispiel dazu:

Es war einmal eine kleine süße Dirne, die hatte jedermann lieb, der sie nur ansah, am allerliebsten aber ihre Großmutter, die wußte gar nicht, was sie alles dem Kinde geben sollte. Einmal schenkte sie ihm ein Käppchen von rotem Sammet,...

Erzeugt wurde es mit:

```
\usepackage{calc}
\noindent\fbox{%
  \begin{minipage}{\linewidth-2\fboxsep-2\fboxrule}%
    Es war einmal eine kleine süße Dirne, die hatte jedermann lieb,
    der sie nur ansah, am allerliebsten aber ihre Großmutter, die
    wußte gar nicht, was sie alles dem Kinde geben sollte. Einmal
    schenkte sie ihm ein Käppchen von rotem Sammet,...
  \end{minipage}%
}
```

Nachteilig an diesem Verfahren ist der Umstand, dass innerhalb einer minipage-Umgebung kein Seitenumbruch stattfindet. In ungünstigen Fällen wandert der gerahmte Text auf die nächste Seite und hinterlässt ein hässliches „Loch“. Abhilfe schafft das Paket framed [1]. Mit ihm gelingen auch Umbrüche innerhalb hervorgehobener Abschnitte wie das nächste Beispiel zeigt:

...und weil ihm das so wohl stand und es nichts anders mehr tragen wollte, hieß es nur das Rotkäppchen. Eines Tages sprach seine Mutter zu ihm: „Komm, Rotkäppchen, da hast du ein Stück Kuchen und eine Flasche Wein, bring das der Großmutter hinaus; sie ist krank und schwach und wird sich daran laben. Mach dich auf, bevor es heiß wird, und wenn du hinauskommst, so geh hübsch sittsam und lauf nicht vom Weg ab, sonst fällst du und zerbrichst das Glas, und die Großmutter hat nichts. Und wenn du in ihre Stube kommst,

so vergiß nicht, guten Morgen zu sagen, und guck nicht erst in alle Ecken herum.“

„Ich will schon alles gut machen“, sagte Rotkäppchen zur Mutter und gab ihr die Hand darauf.

Der zugehörige Quellcode sieht gekürzt wie folgt aus:

```
\usepackage{framed,color}
\definecolor{shadecolor}{gray}{.85}
\begin{shaded}% oder \begin{framed}
  ...und weil ihm das so wohl stand und es nichts anders mehr tragen
  wollte, hieß es nur das Rotkäppchen. Eines Tages sprach seine
  Mutter zu ihm: "‘Komm, Rotkäppchen, da hast du ein Stück Kuchen
  und eine Flasche Wein, bring das der Großmutter hinaus; sie ist...
\end{shaded}% oder \end{framed}
```

Nähere Hinweise zum Paket framed findet man direkt in der Datei framed.sty.

## Literatur

- [1] Donald Arseneau: *The framed Package*; Juli 2001; CTAN: tex-archive/macros/latex/contrib/other/misc/framed.sty.

## Satzspiegelkonstruktionen im Vergleich

oder „Warum sind die Ränder bei KOMA-Script eigentlich so?“

### Markus Kohm

Die Konstruktion von Satzspiegeln, also die Bestimmung des Textbereichs auf der Seite, ist eine Kunst, die seit dem Mittelalter von Profis gepflegt wird. Im Laufe der Zeit wurden viele unterschiedliche Verfahren erdacht, verworfen, vergessen, aber auch weiterentwickelt und dokumentiert. Einige dieser Verfahren sind mit Schlagworten belegt, die sie legendär machen. Obwohl angeblich an jeder Legende auch etwas Wahres ist, birgt die Legendenbildung auch Gefahren. Es gilt daher, Legende und Wirklichkeit miteinander zu vergleichen.

## Einleitung

oder „*Warum ist ein Satzspiegel eigentlich wichtig?*“

Der Satzspiegel ist wie der Rahmen eines Bildes. Ein echter Rembrandt in einem schiefen, bunten, neonfarbenen PVC-Rahmen wird immer wie eine billige Kopie wirken. Ebenso wird ein inhaltlich perfektes Dokument mit verkorkstem Satzspiegel nicht die Geltung erfahren, die es verdient. Obwohl diese Tatsache unstrittig sein sollte, wagen sich Laien immer wieder daran, eigene Satzspiegel zu entwerfen. In einigen Fällen werden dabei Einstellungen aus Textverarbeitungsprogrammen übernommen, die kaum die Bezeichnung „Satzspiegel“ verdienen, geschweige denn als Konstruktion zu bezeichnen sind.

Bei einigen Laien haben sich auch Schlagworte und Teile der damit verbundenen Konstruktionen festgesetzt. Leider sind den Ergebnissen, die dieses Halbwissen zeitigt, oft die nicht beachteten Voraussetzungen der Satzspiegelkonstruktion deutlich anzusehen. Aber auch wenn sie anderen Laien nicht auf den ersten Blick unangenehm auffallen, so können sie unterschwellig einen unerwünschten Eindruck erzeugen. Deshalb folgen nun zunächst einige grundsätzliche Überlegungen zu Satzspiegeln.

## Allen Satzspiegeln gemeinsam

oder „*Welche Rolle spielt Harmonie in der Satzspiegelkonstruktion?*“

Jan Tschichold war unbestreitbar eine der schillerndsten Persönlichkeiten in der Typografie des 20. Jahrhunderts. Dies liegt sicher darin begründet, dass seine typografischen Aufsätze und Abhandlungen eine Sprache verwenden, die man eher in einem Roman als in einem wissenschaftlichen Werk vermuten würde. Einige behaupten, Tschichold sei der Begründer der modernen Typografie. Andere sehen in ihm im Gegenteil einen überholten Verfechter konservativen Denkens. Das Eigenartige am Typografen Tschichold ist, dass er sich anscheinend häufig widerspricht. In Wahrheit jedoch lassen seine Aufsätze bei zeitlicher Einordnung einen lernenden und wachsenden Geist erkennen. Die scheinbaren Widersprüche sind dabei keine sich wandelnden Grundsätze, sondern Neubewertungen an Hand fester Grundsätze.

Diese Grundsätze haben auch in der modernen Typografie ihre Berechtigung, wenn sie auch für den Profi kein zwingendes Korsett darstellen. Für den Laien jedoch haben diese Grundsätze den unschätzbaren Vorteil, dass sie ihm das Leben erleichtern. Wenn die Grundsätze eingehalten sind, ist die Wahrscheinlichkeit, dass das Ergebnis negativ bewertet werden muss, gering. Je mehr die Grundsätze verletzt sind, desto dünner wird das Eis. Während

also der Profi weiß, was er tut, und Verletzungen der Grundsätze bewusst einsetzen kann, sollte der Laie sie meiden wie der Teufel das Weihwasser.

Der wichtigste Grundsatz in der Typografie, von dem viele andere Regeln abgeleitet werden, lautet: Harmonie! Das menschliche Auge oder besser gesagt, die Sinne des Menschen streben ständig in gleicher Weise nach Harmonie wie das Universum nach Chaos strebt.<sup>1</sup> Was Harmonie im Einzelnen bedeutet, ist auch Bestandteil der Kultur. Da es hier um Typografie des Westens geht, bestimmt die westliche Kultur den zugrunde liegenden Harmoniebegriff.

Es stellt sich damit die Frage, wann eine Seite harmonisch ist. Um diese Frage beantworten zu können, müssen zunächst die bestimmenden Größen einer Seite festgestellt werden. Der ungeübte Laie wird sagen: die Breite beziehungsweise Höhe der Ränder oder des Satzspiegels. Genaugenommen hat er damit nicht unrecht. Leider bieten aber die absoluten Größen wenig Möglichkeit einer schematischen Betrachtung. So lässt sich eine Frage, ob ein äußerer Rand von 2 cm besser ist als ein äußerer Rand von 3 cm, kaum beantworten, wenn man nicht weiß, in welcher Beziehung oder welchem *Verhältnis* dieser Rand zu anderen Elementen der Seite steht. Schon allein, um eine Satzspiegelkonstruktion nicht allein für eine Papiergröße anwenden zu können, sind deshalb die Verhältnisse der Elemente und innerhalb der Elemente einer Seite die bestimmenden Größen.

Wir alle kennen aus dem Alltag Verhältnisse, die uns besonders harmonisch erscheinen. So gilt das Seitenverhältnis eines Fernsehgeräts oder Computermonitors mit 4:3 oder eines Breitformatbildes mit 16:9 als harmonisch. Auch andere Verhältnisse mit besonderen Eigenschaften gelten als sehr harmonisch. Der *goldene Schnitt* spielt dabei eine wichtige Rolle. Der goldene Schnitt ist die Lösung des folgenden Problems: Wie muss ein Ganzes geteilt werden, so dass zwei ungleiche Teile entstehen, bei denen das Verhältnis des kleineren Teils zum größeren Teil dem Verhältnis des größeren Teils zum Ganzen entspricht? Tabelle 1 gibt zu einem gegebenen kleinen Teil a jeweils das errechnete größere Teil b und das Ganze c an. In der Typografie werden jedoch in der Regel die ganzzahligen Verhältnisse verwendet, die in Tabelle 2 angegeben sind. Besonders interessant ist hier die fortlaufende Reihe 1:2:3:5:8:13:21:34:55:89... , die erstaunlich genau ist. Wir werden darauf noch zurückkommen. Diese Reihe entspricht übrigens der Fibonacci-

<sup>1</sup>Nebenbei bemerkt gibt es in der Chaosforschung die Theorie, dass das Chaos letztlich zur perfekten Harmonie führt.

a: b: c	a: b: c
1: 1,6: 2,6	1: 2: 3
2: 3,2: 5,2	2: 3: 5
3: 4,9: 7,9	3: 5: 8
4: 6,5: 9,5	4: 6: 10
	5: 8: 13
	6: 10: 16
	7: 11: 18
	8: 13: 21

Tabelle 1: Verhältnisse im goldenen Schnitt

Tabelle 2: Ganzzahlige Verhältnisse im goldenen Schnitt

Reihe, bei der man ein Glied aus der Summe der beiden Vorgänger erhält. Dies ist eine zusätzliche Harmoniebeziehung.

Ein anderes bekanntes, ganzzahliges Verhältnis ist der Villardsche Teilungskanon. Bei diesem wird ein Ganzes so in zwei Teile zerlegt, dass das kleinere Teil halb so groß ist wie das Größere. Es beschreibt also das Verhältnis 1:2. Dieses passt auch zum ganzzahligen goldenen Schnitt.

Neben besonderen Verhältnissen spielt aber auch ein anderer optischer Eindruck eine wichtige Rolle. Betrachten wir einmal Abbildung 1. Der äußere Rahmen soll eine Doppelseite in einem Buch symbolisieren. Die beiden dunkelgrauen Kästen sind die Satzspiegel auf den beiden Seiten. Die hellgrauen Kästen sind Elemente, die in unserem Gehirn unwillkürlich entstehen, obwohl sie auf der Seite normalerweise weiß bleiben. Der Eindruck dieser Kästen entsteht dadurch, dass im Gehirn die Kanten der optischen Ränder des Satzspiegels nach außen verlängert werden. Abhängig von der Größe und Position des Satzspiegels kann dieser Eindruck für einzelne oder alle Kästen stärker oder schwächer sein. Ein besonders harmonischer Eindruck entsteht nun, wenn nicht nur der Satzspiegel dasselbe Seitenverhältnis wie die Seite hat, sondern auch die Kästen A, B und C sowie die quasi liegenden Kästen a, b und c und die ebenfalls liegende Doppelseite. Damit haben praktisch alle optischen Elemente der Seite das gleiche Seitenverhältnis. Ein stärkerer Eindruck von Harmonie ist kaum möglich. Kaum schwächer ist der Eindruck, wenn die Doppelseite und die Kästen a, b und c dasselbe Seitenverhältnis haben, aber dieses Seitenverhältnis vom gemeinsamen Seitenverhältnis der einzelnen Seite, des Satzspiegels und der Kästen A, B und C abweicht. Dies

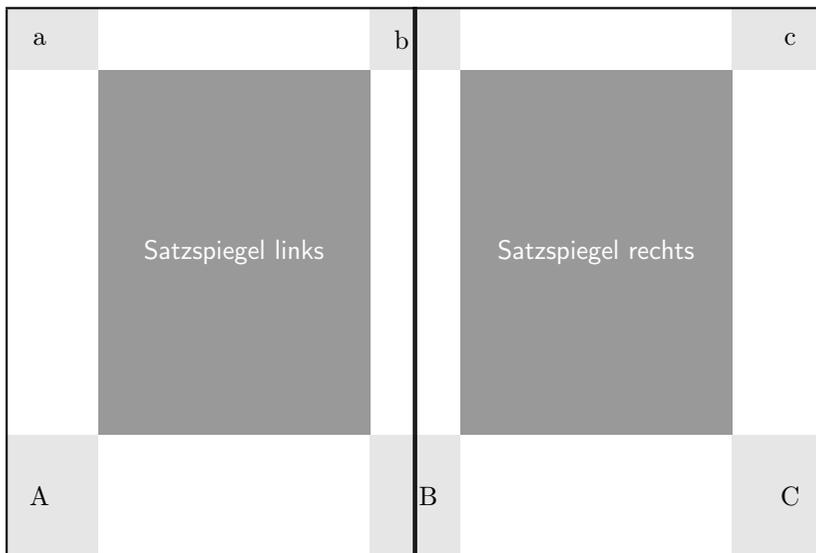


Abbildung 1: Harmonische Größen bei der Satzspiegelkonstruktion

kann wiederum dadurch verstärkt werden, dass die beiden Seitenverhältnisse ihrerseits in einem günstigen Verhältnis zueinander stehen.

Eine interessante Frage ist nun, ob sich die Harmonie besonderer Verhältnisse und die Harmonie der Elemente mit gleichem Seitenverhältnis in Einklang bringen lassen. Ist dies der Fall, dann ist wirklich maximale Harmonie erreicht. Größtmögliche Harmonie lässt gleichzeitig auf eine günstige Satzspiegelkonstruktion schließen. Mindestvoraussetzung der Harmonie ist übrigens – und darüber sind sich alle Typografen einig – die Übereinstimmung des Seitenverhältnisses des Satzspiegels mit dem Seitenverhältnis der Buchseite [3].

Bei der Analyse mittelalterlicher Handschriften erkannte Jan Tschichold übrigens noch eine weitere Harmonie. Er stellte fest, dass bei wertvollen Handschriften der Satzspiegel und – lässt man die kunstvollen Illustrationen weg – der weiße Anteil der Seite ungefähr gleich groß sind. Es ist wenig überraschend, dass Ästheten, Typografen und altgelernte Buchdrucker an einem Großteil der heutigen Bücher vor allem bemängeln, dass die Ränder viel zu schmal und die Satzspiegel viel zu groß sind.

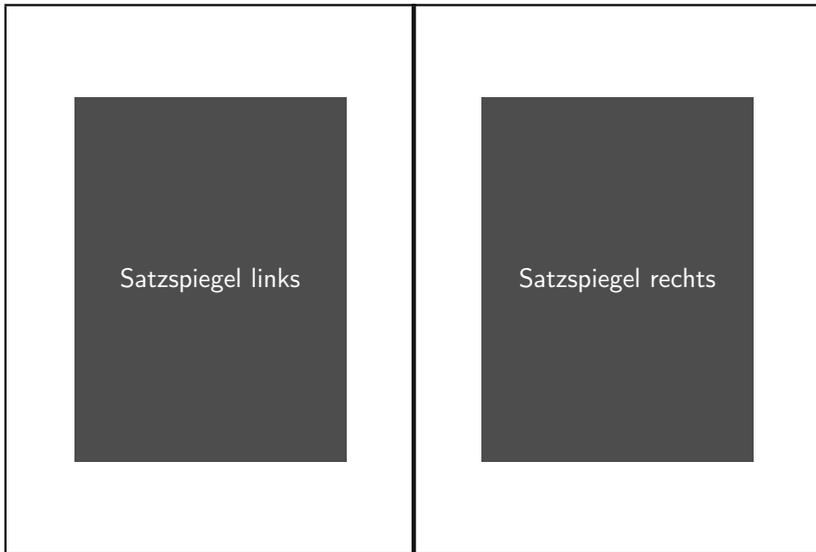


Abbildung 2: Negativbeispiel einer Satzspiegelkonstruktion

Es gibt noch einen Punkt, über den sich alle Typografen einig sind. Der Fußsteg muss immer größer als der Kopfsteg sein. Der Grund dafür ist einfach: Unser Auge sieht die Mitte des Papiers nicht in der Mitte sondern darüber, jedenfalls dann, wenn darauf eine grau wirkende Fläche liegt. Ähnlich verhält es sich mit dem Bundsteg und dem Außensteg. Wie es scheint, ziehen sich die grauen Flächen gegenseitig an. Abbildung 2 zeigt dieselbe Papierfläche und denselben Satzspiegel wie Abbildung 1, allerdings vertikal und horizontal zentriert. In aller Regel werden Sie jedoch den Eindruck haben, dass der Bundsteg breiter ist als der Außensteg und der Kopfsteg breiter als der Fußsteg. Je dunkler die Fläche des Satzspiegels, desto stärker wird dieser Eindruck. Ich versichere jedoch, dass der Eindruck täuscht.

## Vom Mittelalter bis zur Renaissance

oder „Was können wir von den alten Meistern lernen?“

Im Mittelalter wurden Bücher zwar entweder abgeschrieben oder in Reibetechnik gedruckt, aber Typografie gab es damals schon. Zwar war Typografie im Mittelalter weder Handwerk noch Wissenschaft, aber sie wurde von

Schreibern und Druckern gepflegt. Dabei gingen im Idealfall die Fähigkeiten des Meisters auf den Lehrling über und wurden von diesem verbessert. Bekanntlich will die Jugend seit Urzeiten besser als die Alten sein.

Wie bereits im vorigen Kapitel erwähnt, war es im Mittelalter durchaus üblich, dass der Satzspiegel nur ungefähr die Hälfte einer Seite ausmachte. Das wurde dadurch erreicht, dass als Höhe des Satzspiegels oft die Breite des Papiers verwendet wurde. Das verwendete Papier hatte häufig ein Seitenverhältnis von 2:3 oder 3:4. Die Ränder wurden im Verhältnis Bundsteg:Kopfsteg:Außensteg:Fußsteg von 2:3:4:6 gewählt [4]. Aus Tabelle 2 wissen wir, dass dabei das Verhältnis 2:3 des Seitenformats und von Bundsteg:Kopfsteg ebenso im goldenen Schnitt liegt, wie das Verhältnis 1:2 von Bundsteg:Außensteg beziehungsweise Kopfsteg:Fußsteg.

Nehmen wir das Beispiel aus Abbildung 3. Es ergeben sich folgende Verhältnisse:

Papierbreite	: Papierhöhe	= 2 : 3
Satzspiegelbreite	: Satzspiegelhöhe	= 2 : 3
Bundsteg	: Kopfsteg	= 2 : 3
Außensteg	: Fußsteg	= 2 : 3
innerer Rand	: Fußsteg	= 2 : 3
Seitenhöhe	: Doppelseitenbreite	= 3 : 4
Kopfsteg	: Außensteg	= 3 : 4
Bundsteg	: Außensteg	= 1 : 2
Kopfsteg	: Fußsteg	= 1 : 2
Randhöhe	: Satzspiegelhöhe	= 1 : 2
Randbreite	: Satzspiegelbreite	= 1 : 2

Die letzten beiden Verhältnisse wurden bei der Konstruktion bewusst so gewählt. Ich komme später darauf zurück. Wie man sieht, genügen drei Verhältnisse, um die Konstruktion vollständig zu beschreiben.

Die dünnen, schwarzen Linien in Abbildung 3 sind diagonale Hilfslinien. Unabhängig von der Größe des Satzspiegels muss seine äußere obere Ecke immer auf der Diagonalen der Doppelseite und seine innere obere und äußere untere Ecke immer auf der Diagonalen der Einzelseite liegen. Wenn diese Bedingung erfüllt ist, dann ist auch das Verhältnis 2:3:4:6 sichergestellt. Die beiden dünnen Hilfslinien dienen einer Konstruktion, auf die ich noch zurückkomme.

Wie sieht das Ganze nun aus, wenn das Papier nicht das Seitenverhältnis 2:3, sondern 3:4 hat? Wie bereits erwähnt, erscheint uns dieses Seitenver-

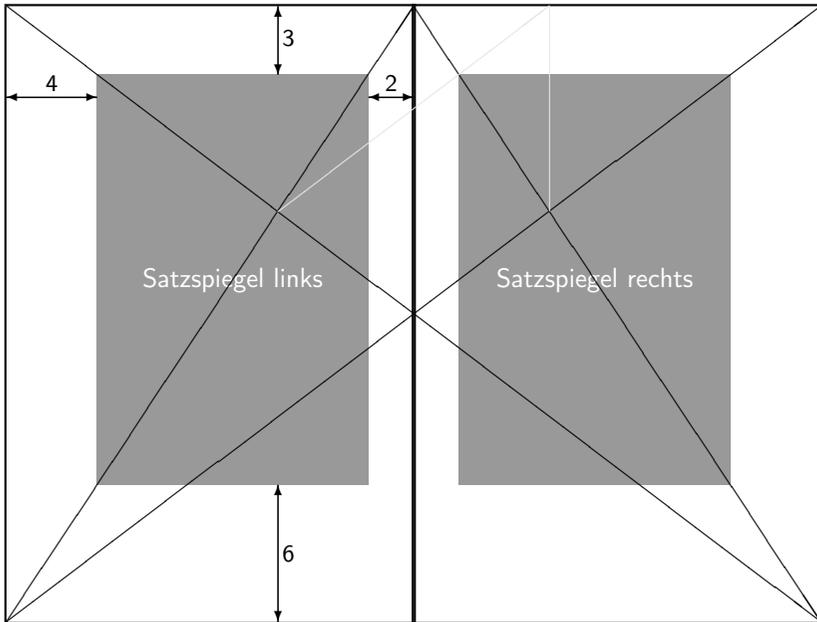


Abbildung 3: Mittelalterliche Satzspiegelkonstruktion mit dem Verhältnis  
 Bundsteg : Kopfsteg : Außensteg : Fußsteg = 2 : 3 : 4 : 6 für Octave-Papier

hältnis ebenfalls harmonisch, auch wenn es im Gegensatz zu 2 : 3 nicht in den goldenen Schnitt passt. Traditionell wird nun das Ränderverhältnis 3 : 4 : 6 : 8 gewählt.

Nehmen wir das Beispiel aus Abbildung 4. Es ergeben sich folgende Verhältnisse:

Papierbreite	: Papierhöhe	= 3 : 4
Satzspiegelbreite	: Satzspiegelhöhe	= 3 : 4
Bundsteg	: Kopfsteg	= 3 : 4
Außensteg	: Fußsteg	= 3 : 4
innerer Rand	: Fußsteg	= 3 : 4
Seitenhöhe	: Doppelseitenbreite	= 2 : 3
Kopfsteg	: Außensteg	= 2 : 3
Bundsteg	: Außensteg	= 1 : 2
Kopfsteg	: Fußsteg	= 1 : 2

Randhöhe	: Satzspiegelhöhe	= 1 : 2
Randbreite	: Satzspiegelbreite	= 1 : 2

Auch dieses wirkt unbestreitbar sehr harmonisch. Und obwohl hier lediglich die Verhältnisse 2 : 3 mit 3 : 4 den Platz getauscht haben, muss man zugeben, dass das Schema von Abbildung 4 noch etwas angenehmer wirkt als das von Abbildung 3. Hier wirken nun nicht mehr allein die Verhältnisse, sondern auch die absoluten Größen. Dies gilt insbesondere für den unteren Rand. Aber auch insgesamt erscheint Octave-Papier gestreckter – ja geradezu härter – als Quart-Papier. Man erlaube mir die Anmerkung, dass ich mit Octave den Vater und mit Quart die Mutter des wohlgerateten Layouts verbinde.

In der Renaissance erkannte man den langen, nackten Hals bei Octave offenbar auch. Man versuchte sogleich mit einem einfachen Kunstgriff dieses Problem zu beseitigen, nämlich der Änderung des Ränderverhältnisses von 2 : 3 : 4 : 6 in 2 : 3 : 4 : 5. Abbildung 5 zeigt die Auswirkungen. Dem geübten Auge und dem aufmerksamen Leser wird sofort auffallen, dass dieses Verhältnis einige Nachteile mit sich bringt. Betrachten wir die Verhältnisse etwas genauer:

Papierbreite	: Papierhöhe	= 2 : 3
Satzspiegelbreite	: Satzspiegelhöhe	= 27 : 41
Bundsteg	: Kopfsteg	= 2 : 3
Außensteg	: Fußsteg	= 4 : 5
innerer Rand	: Fußsteg	= 4 : 5
Seitenhöhe	: Doppelseitenbreite	= 3 : 4
Kopfsteg	: Außensteg	= 3 : 4
Bundsteg	: Außensteg	= 1 : 2
Kopfsteg	: Fußsteg	= 3 : 5
Randhöhe	: Satzspiegelhöhe	= 8 : 19
Randbreite	: Satzspiegelbreite	= 1 : 2

Negativ fällt auf, dass das Seitenverhältnis des Satzspiegels nicht mehr mit dem Seitenverhältnis des Papiers übereinstimmt. In der Konstruktion ist das leicht daran zu erkennen, dass die linke untere Ecke der linken Seite die Diagonale verlässt. Des Weiteren stehen Bundsteg und Außensteg mit 1 : 2 in einem anderen Verhältnis zueinander als Kopfsteg und Fußsteg mit 3 : 5. Vorteilhaft ist dabei nur, dass beide Verhältnisse dem ganzzahligen goldenen Schnitt entsprechen. Aus dem verletzten Seitenverhältnis resultiert dann natürlich auch, dass die horizontalen Ränder in einem anderen Verhältnis zum

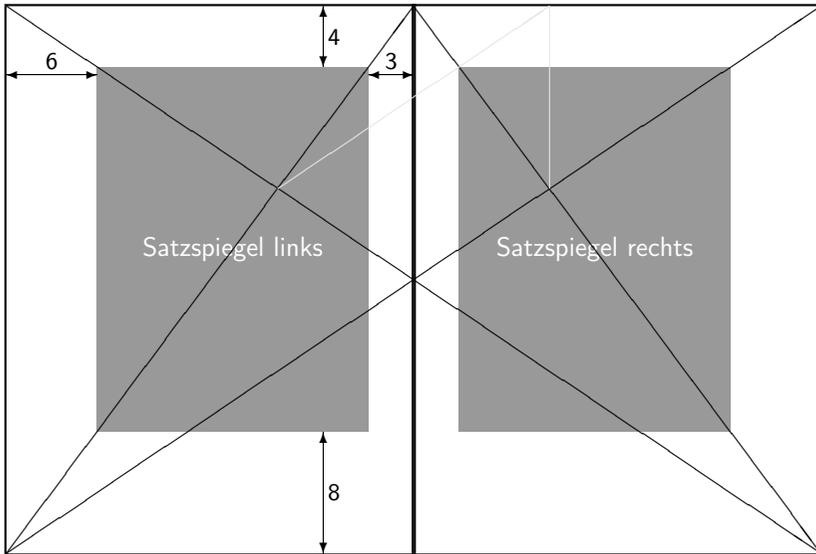


Abbildung 4: Mittelalterliche Satzspiegelkonstruktion mit dem Verhältnis  
 Bundsteg : Kopfsteg : Außensteg : Fußsteg = 3 : 4 : 6 : 8 für Quart-Papier

Satzspiegel stehen als die vertikalen. Bei idealer Konstruktion wie in den beiden Abbildungen gezeigt, wird dann aus 1 : 2 plötzlich 8 : 19. Dieses Verhältnis passt auch nicht so recht in eines der harmonischen Teilungsverhältnisse. Insgesamt ergibt sich nun ein Wirrwarr vieler verschiedener Verhältnisse. Dies ist ein guter Hinweis auf eine gestörte Harmonie.

Einige Typografen lehnen die Renaissance-Konstruktion schon allein wegen des Verstoßes beim Satzspiegelverhältnis ab. Allerdings liegt die Abweichung bei weniger als einem Prozent. Mit Feingefühl kann ein Typograf selbst damit arbeiten. Dem Laien werden jedoch die wenigsten Typografen diese Konstruktion empfehlen, denn in der Praxis können zu dieser konstruktionsbedingten Abweichung weitere Abweichungen kommen.

In der Literatur konnte ich keinen Hinweis darauf finden, dass in der Renaissance diese Abweichung auch auf Quart-Papier übertragen wurde. Ich verzichte hier deshalb ebenfalls darauf, dies zu behandeln. Erwähnt sei jedoch, dass eine ähnliche Abweichung bei Quart beispielsweise durch das Verhältnis 3 : 4 : 6 : 7 erreicht werden könnte.

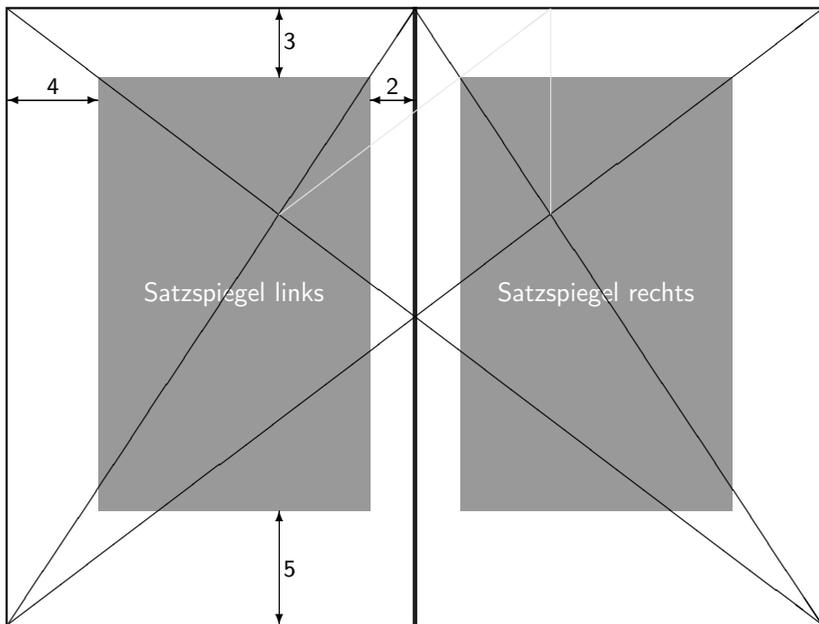


Abbildung 5: Renaissance-Satzspiegelkonstruktion mit dem Verhältnis  
 Bundsteg : Kopfsteg : Außensteg : Fußsteg = 2 : 3 : 4 : 5 für Octave-Papier

## Die Moderne

oder „Was hat es mit den Hilfslinien auf sich?“

In der Bauhauszeit wurde Ästhetik und die Suche danach zum Prinzip erklärt. Man mag zu dem, was diese Epoche als ästhetisch hervorgebracht hat, stehen wie man will, die Typografie erhielt in dieser Zeit neue und wiederentdeckte Impulse. Für die Satzspiegelkonstruktion von Bedeutung ist, dass zu dieser Zeit die Papiere der DIN-Reihen A, B, C und D bereits Verbreitung gefunden hatten. Es erschien wünschenswert, auf Konstruktionen zurückzugreifen, die unabhängig vom Papierformat funktionieren.

Vermutlich vom holländischen Typografen van de Graaf stammt die Konstruktion, die bereits in den Abbildungen der vorherigen Abschnitte skizziert ist [5]. Man legt dabei zunächst in jede Einzelseite die Diagonale von außen unten nach innen oben. Dann fügt man die Diagonalen der Doppelseite hinzu.

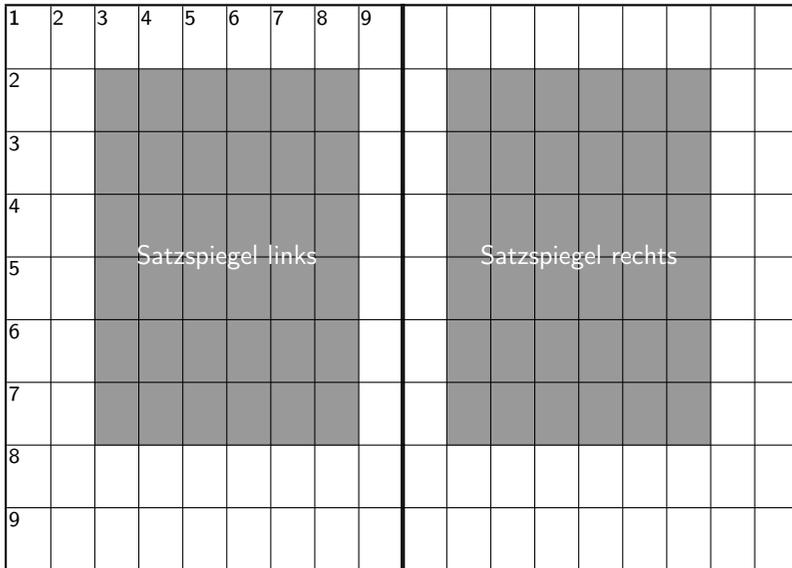


Abbildung 6: Neunerteilung am Beispiel von Papier der DIN-A-Reihe

Vom Schnittpunkt der ansteigenden Doppelseitendiagonalen mit der Diagonalen der rechten Seite geht man senkrecht nach oben. Von diesem Punkt aus zieht man eine Linie zum Schnittpunkt der abfallenden Doppelseitendiagonalen mit der Diagonalen der linken Seite. Der Schnittpunkt dieser Linie mit der Diagonalen der rechten Seite ist die linke obere Ecke des Satzspiegels der rechten Seite. Die rechte obere Ecke des Satzspiegels ergibt sich dann aus dem Schnittpunkt einer horizontalen Linie von der linken oberen Ecke des Satzspiegels mit der ansteigenden Doppelseitendiagonalen. Der Schnittpunkt einer senkrechten Linie von dieser Ecke mit der Seitendiagonalen bringt dann die rechte untere Ecke. Die vierte Ecke ergibt sich durch die rechten Winkel des Satzspiegels von allein.

Jan Tschichold erkannte, dass diese Konstruktion genau zu der klassischen Einteilung führt, die in den Abbildungen 3 und 4 bereits zu sehen war [7]. Des Weiteren entspricht sie einer Teilung der Papiervertikalen und der Papierhorizontalen in jeweils neun Teile wie sie in Abbildung 6 zu sehen ist. Sie ist deshalb auch als Neunerteilung oder klassische Neunerteilung in der

Literatur zu finden. Diese Satzspiegelkonstruktion wurde vom Mittelalter bis in unser Jahrhundert für Literatur von Cicero bis zu Minnegesängen verwendet. Im Gegensatz zu festen Ränderverhältnissen funktioniert sie unabhängig vom Papierformat und führt dann zu den passenden Ränderverhältnissen. Unverwöhnten – ja! gequälten – Augen erscheinen die Ränder außerordentlich extravaganter. Dem verwöhnten Auge erscheint die Seite überaus elegant. Kritische Leser schreien verzückt auf, haben sie doch endlich Platz für ihre Anmerkungen.

Doch genug der Verzückung! Einige moderne Typografen bezweifeln, dass diese Konstruktion für lebende Kolummentitel wirklich geeignet ist [5]. Seien wir ehrlich: Der Platz, der für Text nach Abzug der Kolummentitel bleibt, schwindet dabei. Lebende Kolummentitel sind darüber hinaus ein Merkmal von wissenschaftlichen Werken, in denen auch häufig Konsultationselemente wie Tabellen und Abbildungen zu finden sind. Auch für diese wird oft viel Platz in der Höhe und auch in der Breite benötigt. Es ist daher wenig verwunderlich, dass schon Jan Tschichold im letzten Jahrhundert erkannte, dass für technische oder allgemeiner für wissenschaftliche Bücher eine Erweiterung gefunden werden muss. Er fand sie einfach, indem er eine Abweichung von der Verwendung von neun Teilen erlaubte und zum  $n$ -mal- $n$ -Raster überging. Daraus resultiert die Rasterkonstruktion.

Bei der Rasterkonstruktion wird die Seite wie bei der Neunerteilung vertikal und horizontal in ein Raster eingeteilt. Dabei werden vertikal und horizontal gleich viele Teile verwendet. Dadurch wird erreicht, dass jedes Kästchen des Rasters dasselbe Seitenverhältnis besitzt wie die Seite. Für den Bundsteg wird eine Spalte, für den Kopfsteg eine Zeile, für den Außensteg zwei Spalten und für den Fußsteg zwei Zeilen des Rasters verwendet. Der Rest ist dann der Satzspiegel. Der einzige Unterschied zur Neunerteilung besteht damit in der Anzahl der Zeilen und Spalten, die verwendet werden. Die Neunerteilung ist also lediglich ein Sonderfall der Rasterkonstruktion.

Die Besonderheit der Rasterkonstruktion besteht darin, dass sie nicht mehr vom Papierformat abhängig ist. Das liegt daran, dass die Verhältnisse Bundsteg:Kopfsteg und Kopfsteg:Außensteg automatisch dem Seitenverhältnis von Satzspiegel und Seite folgen, also die Kästchen A, B, C und a, b, c aus Abbildung 1 automatisch harmonisch sind.

Bedeutet dies nun, dass die Rasterkonstruktion immer ein perfektes Layout liefert? Diese Frage kann mit einem klaren Ja beantwortet werden. Zunächst sollte das Verhältnis Rand:Satzspiegelhöhe beziehungsweise

Rand: Satzspiegelbreite in aller Regel nicht größer als 1:2 werden. Sonst werden die Ränder eindeutig zu groß. Umgekehrt sollte der kleinste sichtbare Rand nie kleiner als 12,5 mm sein. Dieser Wert erscheint willkürlich und ist es teilweise auch. Dieser Vorwurf trifft allerdings mehr oder weniger für alle Erfahrungswerte zu. Er sollte jedoch nicht als Ausrede dafür dienen, Erfahrungswerte zu ignorieren. Der kleinste sichtbare Rand ist in der Regel der Kopfsteg, da der Bundsteg auf Doppelseiten ja doppelt zu sehen ist. Ein kleinster Rand von 19 mm gilt als nahezu optimal und 25 mm sind elegant. Vermutlich hat Letzteres auch irgendwann ein Programmierer gelesen und dann in seine Textverarbeitung als Voreinstellung eingebaut – leider für alle Ränder und leider wurde nie ein Typograf gefragt, ob das so korrekt ist.

Doch die Wahl minimaler oder maximaler Ränder ist nicht das einzige Kriterium. Ebenso wichtig wie ausreichende und nicht übermäßige Randbreiten ist die richtige Breite des Satzspiegels. Hier spielt jedoch der Wert in einem objektiven Maß zunächst eine untergeordnete Rolle. Zwar hat das menschliche Auge klar ein Problem, wenn es vom Ende einer 1 m langen Zeile an deren Anfang zurück soll. So lange man den Kopf beim Lesen aber nicht drehen muss – obwohl man das in der Regel trotzdem macht – ist das Problem der reinen Wegstrecke ein zweitrangiges. Wichtiger ist die Anzahl der Zeichen und die Anzahl der Wörter in einer Zeile. Geübte Leser haben bei gut gewählter Schrift mit ordentlichem, aber nicht übertriebenem Durchschuss kein Problem beim Verfolgen und Zurückspringen, wenn eine Zeile durchschnittlich nicht mehr als 74 Zeichen hat. Angenehm und für ungeübte Leser wünschenswert sind jedoch 62 bis 66 Zeichen. Da das Auge in Wirklichkeit nicht von Buchstabe zu Buchstabe, sondern von Wort zu Wort springt, sollten im allgemeinen nicht mehr als 10 bis 12 Wörter in jeder Zeile stehen. Das begünstigt scheinbar Sprachen mit langen Wörtern oder Wortgefügen. Dem ist nicht so, da das Auge dann wieder nicht mehr von Wort zu Wort, sondern von Teilwort zu Teilwort springt. Dabei spielen Buchstaben mit durchgehenden vertikalen Kanten eine große Rolle. Aber das ist ein anderes Thema.

Das Vorgehen sollte also so sein, dass man zunächst ausmisst, wie breit der Satzspiegel bei der gewählten Schrift maximal sein darf. Dann sucht man nach einem Raster, das zu dieser Breite passt. Als Letztes überprüft man, ob die Bedingungen bezüglich minimaler und maximaler Ränder eingehalten sind. Wenn die Ränder zu schmal sind, wählt man ein größeres Raster, denn kürzere Zeilen im Satzspiegel sind unproblematisch. Was aber tun, wenn die Ränder zu breit sind? Als Notlösung kann man den Satzspiegel etwas breiter wählen und dafür den Durchschuss erhöhen. Dieses Vorgehen ist aber nur

sehr beschränkt anwendbar, etwa bis zu Zeilenlängen von 76 Zeichen und einem Grundlinienabstand von Faktor 1,35 der Schriftgröße. Dabei darf die Schrift auch nicht zu leicht und nicht zu fett sein. Anderenfalls wird der Graueindruck der Seite zu streifig. Die beste Lösung besteht dann darin, entweder das Papierformat oder die Schrift zu wechseln.

Vorteilhaft an der Rasterkonstruktion ist, dass sie sehr einfach mit Computerunterstützung umzusetzen ist. Bei KOMA-Script [2] wird genau dies mit dem Paket `typearea` erreicht. Bei Verwendung der Option `DIVcalc` beachtet das Paket ebenfalls Minimalränder. Dabei werden aber als absolutes Minimum nicht die oben erwähnten 12,5 mm, sondern nur 5 mm verwendet.

## Seltene Alternativen

oder „Wie kann der goldene Schnitt an Glanz verlieren?“

Wie wir bisher gesehen haben, findet der goldene Schnitt in der einen oder anderen Form in vielen klassischen Satzspiegelkonstruktionen Eingang. Bei der Rasterkonstruktion ist das letztlich nur in Form des Verhältnisses 1 : 2 der Fall, das quasi die unterste Stufe des ganzzahligen goldenen Schnitts bedeutet, der Fall und entspricht gleichzeitig dem Villardschen Teilungskanon. Dieses Verhältnis findet sich bei der Rasterkonstruktion wie auch bei den anderen vorgestellten Konstruktionen als Verhältnis von Bundsteg : Außensteg und eingeschränkt Kopfsteg : Fußsteg. Bei der Neunterteilung findet es sich ebenfalls beim Verhältnis der Anzahl der Zeilen beziehungsweise Spalten, die für den Rand verwendet werden, zu denen, die für den Satzspiegel verwendet werden.

Manchmal liest man in der Literatur, dass der goldene Schnitt noch einen Schritt weiter in die Satzspiegelkonstruktion Eingang findet. Dann wird als Verhältnis der Ränder 3 : 5 : 8 : 13 verwendet. Dies ist ein Vier-Zahlen-Ausschnitt aus der oben erwähnten Fibonacci-Reihe. Ein erstes Problem dabei ist, dass das Verhältnis von Bundsteg : Außensteg nun nicht mehr 1 : 2, sondern 3 : 8 ist. Das bedeutet, dass der gesamte innere Rand kleiner als jeder Äußere Rand ist. Dies ist eine Störung der Harmonie [3]. Immerhin ist dabei der innere Rand 25 Prozent kleiner als jeder Äußere. Dazu kommt, dass das Papier in der Mitte von Büchern häufig gewölbt ist und auch damit optisch eine zusätzliche Verkleinerung des Bundstegs eintreten kann, falls diese bei der Konstruktion nicht kompensiert wird. Ein weiteres Problem besteht darin, dass das Verhältnis Bundsteg : Kopfsteg nicht dem Verhältnis Außensteg : Fußsteg entspricht. Die Kästen A und a aus Abbildung 1 haben

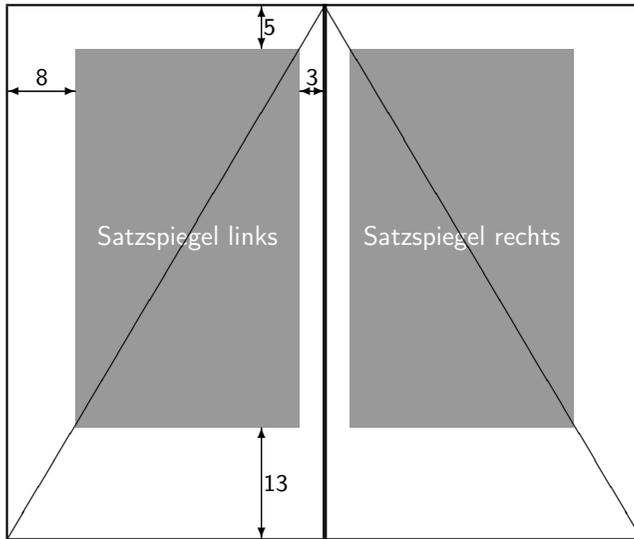


Abbildung 7: Goldener Schnitt für Papier und Satzspiegelkonstruktion

also unterschiedliche Seitenverhältnisse. Die Abweichung fällt mit etwa 2,5 Prozent aber tolerierbar gering aus und ist letztlich darin begründet, dass der ganzzahlige goldene Schnitt eben nicht wirklich der goldene Schnitt ist. Dies gilt dann auch für die Möglichkeit, dass der Satzspiegel das Seitenverhältnis der Seite selbst besitzt. Wie inzwischen häufig erwähnt wurde, ist letzteres für die Harmonie sehr wichtig. Stellt sich also die Frage, wann dies beim goldenen Schnitt der Fall ist. Analog zu den Ränderverhältnissen  $2:3:4:6$  für Octave- und  $3:4:6:8$  für Quart-Papier sollte beim goldenen Schnitt mit dem Ränderverhältnis  $3:5:8:13$  das Seitenverhältnis des Papiers dem goldenen Schnitt, also  $3:5$  oder  $8:13$ , entsprechen. Abbildung 7 zeigt ein Beispiel dafür. Abbildung 8 zeigt hingegen, was passiert, wenn man diese Konstruktion auf Papier der DIN-A-Reihe anwendet. In beiden Abbildungen wirkt nach meiner persönlichen Auffassung der Satzspiegel zu weit nach innen gedrängt. Die vertikale Ausrichtung erscheint eher brauchbar. Dies gilt umso mehr, wenn mit Kolummentiteln gearbeitet wird. In Abbildung 8 fällt das nicht passende Seitenverhältnis des Satzspiegels allerdings bereits deutlich auf. Für das ungeübte Auge habe ich den Eindruck durch eine Hilfslinie noch etwas verdeutlicht. Die Konstruktion ist für Papier der DIN-A-Reihe



Überhaupt sollen möglichst wenig unterschiedliche Verhältnisse auftreten. Letztes Kriterium für den Laien ist dann, ob die Zeilenlänge günstig und der Graueindruck einheitlich ist. Geringe Abweichungen bei der Höhe des Satzspiegels und der Höhe des unteren Randes sollte der Laie tolerieren. Ebenso sollte er bei der Kontruktion die Höhe des unteren Randes eher ab- und die Höhe des Satzspiegels eher aufrunden, wenn es darum geht, Nebenbedingungen einzuhalten. Eine Nebenbedingung bei L<sup>A</sup>T<sub>E</sub>X ist, dass `\textheight` abzüglich `\topskip` ein ganzzahliges Vielfaches von `\baselineskip` sein sollte. Dadurch muss bei einer ganz mit einem Absatz gefüllten Seite der Zeilenabstand nicht gedehnt werden.

Der Anwender kann sich das Leben auch erleichtern und das `typearea`-Paket von KOMA-Script verwenden. Bei Verwendung der Option `DIVcalc` muss er dort in aller Regel auf gar nichts achten. Bei Vorgabe eines eigenen DIV-Wertes, also bei eigener Wahl des Teilungsfaktors für die Rasterkonstruktion, muss er lediglich die Zeilenlänge überprüfen, wobei `typearea` auch hier Unterstützung in Form einer Bewertung des Ergebnisses bietet.

Abschließend sei noch auf ein Zitat aus [6] zurückgegriffen:

Heute sind Milchsack und Tschichold fast vergessen. Jeder Computerbenutzer kann willkürlich ein Buch gestalten. Daher sind die meisten neuen Bücher hässlich. Grundsätzlich haben unsere Bücher folgende Mängel: Der Satzspiegel ist zu groß, die Verlage nutzen das Papier soweit wie nur möglich aus [...] Die Ränderverhältnisse sind zufällig, oft stehen die Kolumnen zu tief.

Ich hoffe, das notwendige Rüstzeug vermittelt zu haben, damit nun auch Computerbenutzer ihre Bücher weniger willkürlich gestalten und auf Ränder mehr Wert legen.

## Glossar

oder „Was versteht man eigentlich unter ...“

In diesem Glossar werden die wichtigsten typografischen Begriffe erklärt, die in diesem Artikel Verwendung finden. Ich habe mir dabei die Freiheit genommen, nicht die Erklärungen aus irgendwelchen Lexika abzuschreiben, sondern meine eigenen Erklärungen und Kommentare abzugeben. Wenn in den Erklärungen ein *logischer Ort* erwähnt wird, so ist damit ein Ort gemeint, der zum einen mit einem einzigen Begriff bezeichnet werden kann, zum anderen auf linken und rechten Seiten symmetrisch zum Satzspiegel ausgerichtet ist.

Beispiele für logische Orte sind: „oben, innen“, „unten, außen“, „mittig, außen“ usw.

*Außensteg* Der Außensteg ist der Bereich zwischen der rechten oder linken, äußeren Kante der Seite und dem *Satzspiegel*. Er wird auch als *äußerer Rand* bezeichnet.

*Bundsteg* Der Bundsteg ist der innere Rand einer Seite. Der Bundsteg der linken Seite bildet mit dem Bundsteg der rechten Seite den inneren Rand. Ob ein Teil des Bundstegs bei der Bindung also durch den Bund verloren geht, oder ob dieser Verlust durch den übergroßen Papierbogen zu kompensieren ist, ist strittig. Unstrittig ist jedoch, dass optische und harmonische Eigenschaften des Bundstegs nur für den sichtbaren Bundsteg angegeben werden können. In diesem Artikel ist daher der Bundsteg mit dem sichtbaren Bundsteg gleichgesetzt.

*Fußsteg* Der Fußsteg ist der Bereich zwischen der unteren Kante der Seite und dem *Satzspiegel*. Er wird auch als *unterer Rand* bezeichnet.

*Kolumnentitel* Man unterscheidet zwischen *lebendem* und *totem* Kolumnentitel. Der *tote Kolumnentitel* ist normalerweise identisch auf jeder Seite am gleichen logischen Ort zu finden. Er ist nicht Teil des *Satzspiegels*. In der Regel ist der tote Kolumnentitel im oberen oder unteren Rand positioniert, wird klein gesetzt und sollte kurz sein. In jedem Fall sollte ein toter Kolumnentitel optisch unauffällig bleiben und so die Harmonie von Satzspiegel und Rändern nicht stören.

Der *lebende Kolumnentitel* wiederholt Elemente der Gliederung des Dokuments. Er ist Teil des Satzspiegels. In der Regel werden Kapitel- und Abschnittsüberschriften für den lebenden Kolumnentitel verwendet. Er ist auf jeder normalen Textseite am gleichen logischen Ort zu finden. Auf besonderen Seiten, beispielsweise wenn Kolumnentitel und Überschrift mit identischem Text untereinander stehen würden, entfällt der lebende Kolumnentitel. Üblich ist, dass der lebende Kolumnentitel in der ersten Zeile des Satzspiegels positioniert wird.

*Kopfsteg* Der Kopfsteg ist der Bereich zwischen der oberen Kante der Seite und dem *Satzspiegel*. Er wird auch als *oberer Rand* bezeichnet.

*Marginalien* Die Marginalien sind Hinweise, Symbole, Ergänzungen und ähnliche Elemente, die neben dem eigentlichen Text angebracht sind und diesen in der Regel unterstützen. Dünn besetzte Marginalien sind solche, die

nur vereinzelt auftreten und optisch nur wenig Platz beanspruchen. Sie befinden sich in der Regel im *Außensteg*, selten im *Bundsteg*. Der *Bundsteg* wird zum einen aufgrund seiner Wölbung ausgespart, zum anderen weil er beim Blättern im Buch nicht ins Auge fällt. Damit können Marginalien im *Bundsteg* nur eine geringere Wirkung hervorrufen.

Dicht besetzte Marginalien sind solche, die gehäuft auftreten oder optisch sehr viel Platz beanspruchen. Sie würden den optischen Eindruck des Randes und damit die Harmonie der Seite stören. Deshalb sind sie ganz oder teilweise Bestandteil des *Satzspiegels*.

*Octave* Octave ist ein Papierformat mit dem Seitenverhältnis 2 : 3.

*Paginierung* Die Paginierung ist die Ausgabe und Positionierung der Seitennummer auf der Seite. Es ist durchaus möglich und statthaft, ein Layout zu entwerfen, bei dem auf einer Seite die Seitennummer mehrfach ausgegeben wird. Allerdings sollte die Seitennummer innerhalb eines Dokuments nicht ihren logischen Ort wechseln. Nur so findet der Leser die Seitennummer leicht. Eine Seitennummer, die der Leser nicht leicht findet, ist nutzlos. Die Paginierung ist nicht Teil des *Satzspiegels*. Es ist jedoch möglich, die Paginierung als Teil des *lebenden Kolumnentitels* innerhalb des Satzspiegels zu positionieren.

*Quart* Quart ist ein Papierformat mit dem Seitenverhältnis 3 : 4.

*Satzspiegel* Unter dem Satzspiegel versteht man die gedachte graue Fläche, die sich ergibt, wenn man um den bedruckten Textbereich einen Rahmen zieht. Paginierung, tote Kolumnentitel und dünn besetzte Marginalien gehören nicht zum Satzspiegel.

Der Laie denkt oft, dass diese Erklärung den Satzspiegel von den Rändern loslöst, so dass er einen einmal gewählten Satzspiegel frei auf der Seite positionieren kann. Das ist falsch, denn ein Satzspiegel sollte immer Ergebnis einer schematischen Konstruktion sein. Nur so sind wiederholbare und für Laien bewertbare Ergebnisse möglich. Damit ist aber auch die Position des Satzspiegels festgelegt.

## Literatur

oder „Wo finde ich mehr zu dem Thema?“

- [1] Markus Kohm: KOMA-SCRIPT – *Eine Alternative zu den Standardklassen?*; *Die T<sub>E</sub>Xnische Komödie*; 2/96, S. 14–33; 1996.

- [2] Frank Neukam, Markus Kohm und Axel Kielhorn: *KOMA-Script*; L<sup>A</sup>T<sub>E</sub>X-Paket in Version 2.9j; Sept. 2002; CTAN: [tex-archive/macros/latex/contrib/supported/koma-script/](http://tex-archive/macros/latex/contrib/supported/koma-script/).
- [3] Axel Reichert: *Typografie – Gestaltung einer Beispielklasse*; Tutorium auf der Tagung DANTE'99; CTAN: [info/german/typografie/](http://info/german/typografie/).
- [4] Hermann Rotermund: *Gute Typographie auf Papier und im Web*; URL: <http://www.weisses-rauschen.de/hero/Typographie.htm>.
- [5] Jürgen F. Schopp: *Blätter (und Tipps) zur Typografie*; 2. Version; Sept. 2002; URL: <http://www.uta.fi/~trjusc/typoblatt.htm>.
- [6] Martin Z. Schröder: *Die Anmut des Unscheinbaren*; Berliner Zeitung von 10. Oktober 1998; URL: [http://www.berlinonline.de/wissen/berliner\\_zeitung/archiv/1998/1010/magazin/0003/](http://www.berlinonline.de/wissen/berliner_zeitung/archiv/1998/1010/magazin/0003/).
- [7] Jan Tschichold: *Willkürfreie Maßverhältnisse der Buchseite und des Satzspiegels*; in *Schriften 1925–1974*; Brinkmann & Bose; Berlin; 1992.

## Aus alt mach neu – METAPOST Recycling

Jens-Uwe Morawski

METAPOST ist durch seinen vielfältigen Funktionsvorrat und durch seine gute Integration in das T<sub>E</sub>X-System ein gern genutztes Werkzeug für ansprechende Illustrationen. Dieser Artikel zeigt, wie mit geringem Aufwand bereits vorhandene METAPOST-Arbeiten in eine neue Grafik integriert werden können.

Wie war das gleich nochmal. . .

So, oder so ähnlich, beginnt der Satz mit dem man sich auf die Suche nach einer alten METAPOST-Datei begibt. Dabei ist man auf der Suche nach einer Lösung für ein aktuelles Problem und weiß, dass irgendwo in den vielen Dateien die Perle mit entsprechendem METAPOST-Code bereits steckt.

Ein kurzer Testlauf bestätigt: Ja, genau, Grafik 4 enthält das Gesuchte! Aber oje, die lange stiefmütterliche Behandlung hat der Datei nicht gut getan – der damals noch so elegante Code ist absolut unverständlich, und irgendwie hatten die eingestreuten Kommentare damals auch eine sinnvollere Aussage. Was tun? Man will doch nur das, was in Grafik 4 definiert ist, in die neue Datei übernehmen!

Wie das geht, wird hier gezeigt und ist auch geeignet, um sich beispielsweise eine kleine Grafikkbibliothek aufzubauen.

## Die Perle

Die in den folgenden Abschnitten gezeigten Erklärungen und Definitionen sind zwar allgemein gültig, trotzdem möchte ich hier kurz abschweifen und genauer auf die Datei eingehen, die die gesuchte Grafik enthält. Ich mache das aus dem Grund, da hier noch ein paar Anregungen zu finden sind, die im Zusammenhang mit Gerd Neugebauers Artikel [7] stehen.

Die wiedergefundene Perle, also die Datei mit der einzulesenden Grafik, soll im Folgenden `dpts.mp` heißen. Ihr Aufbau gliedert sich in sechs Teile. Im ersten wird ein Variablen-Array vom Typ *path* mit Pfadangaben gefüllt. Im zweiten Teil werden aus den Pfaden des Arrays geschlossene Kurvenzüge gebildet – vier an der Zahl. In den folgenden vier Teilen wird jeweils eine METAPOST-Grafik definiert, wobei immer einer der vier zusammengesetzten Pfade mit einer Farbe gefüllt wird. Zugegeben, diese METAPOST-Datei ist sehr einfach und entspricht somit nicht unbedingt dem Szenario der Einführung. Aber hier soll mehr das *Wie* und *Wieso* so der Entstehung interessieren.

Vereint man alle vier Grafiken zu einer und stellt auch noch die umschließenden Pfade dar, dann ergibt sich die in Abbildung 1 dargestellte Grafik. Die markierten Punkte sind zusätzlich eingefügt und interessieren uns erst weiter unten. Unschwer ist zu erkennen: die Abbildung stellt die Umrisse der Länder Deutschland, Polen, Tschechische Republik und Slowakei dar.

Die Pfade für die Grafiken wurden ähnlich dem von Gerd Neugebauer beschriebenen Verfahren ermittelt, siehe [7]. Im Gegensatz zu ihm bin ich nicht durch Scannen sondern durch Download von [8] zur grundlegenden Pixel-Grafik gekommen. Viel entscheidender ist aber, dass ich zum Nachzeichnen der Grenzlinien nicht von einem geschlossenen Kurvenzug, beispielsweise einem Dreieck, sondern von einem offenen ausgegangen bin. Der Grund für dieses Vorgehen liegt natürlich in der unterschiedlichen Problemstellung, da



Abbildung 1: Die vier Grafiken zusammengesetzt

ich mehrere Länder mit teilweise überschneidenden Grenzlinien in eine Vektorgrafik umzuwandeln hatte.

Solch ein offener Kurvenzug wurde immer dort gezeichnet, wo eine Grenzlinie ohne Abzweigung vorliegt. Würde man von einem geschlossenen Pfad ausgehen, dann hätte man das Problem, dass Pfadabschnitte doppelt bearbeitet werden und sich auch noch exakt decken müssten.

Damit ergeben sich neun einzelne, offene Kurvenzüge, die an den in Abbildung 1 mit Punkten markierten Stellen zusammenstoßen. Damit Anfangs- und Endpunkte auch exakt aufeinander zu liegen kommen, kann man sich meist der Hilfsmittel der Grafikprogramme bedienen. Im vorliegenden Fall wurde *SodiPodi* [5] benutzt, weil es nicht nur Rasterfang von Punkten sondern auch frei definierbare Fanglinien unterstützt.

Nachdem die neun Pfade gezeichnet waren, habe ich das Pixelbild entfernt und eine PostScript-Datei erzeugt. Diese Datei wurde nachfolgend mit *ps-to-edit* [2] in METAPOST umgewandelt.

Die dabei erzeugte METAPOST-Datei sieht ausschnittsweise wie folgt aus. Da die Pfadspezifikation recht lang werden kann, habe ich sie hier zur Darstellung verkürzt. Dies ist mit ... angedeutet, sollte somit nicht mit dem gleichlautenden METAPOST Pfadbefehl verwechselt werden.

```

beginfig(1);
pickup pencircle scaled 1bp;
draw (175.918,232.378)--(165.152,239.015)
    ...--(132.008,237.798)--(177.688,57.7546);
drawoptions (withcolor (1,0,0));
draw (175.918,232.378)--(175.918,227.982)
    ...--(197.792,145.395)--(197.614,137.815);
endfig;

```

Die gesuchten Pfade finden sich als Argument zu den `draw`-Anweisungen in der Datei. Da mich nur diese Pfade interessierten, habe ich im ersten Schritt alles andere gelöscht. Nachfolgend wurden die Pfaddefinitionen dem oben bereits genannten Array vom Typ *path* zugewiesen. Damit sind die neun Pfade durch die Variablen `dpts_map[1]` bis `dpts_map[9]` definiert.

```

path dpts_map[] ;

dpts_map[1] := (175.918,232.378)--(165.152,239.015)
    ...--(132.008,237.798)--(177.688,57.7546);

dpts_map[2] := (175.918,232.378)--(175.918,227.982)
    ...--(197.792,145.395)--(197.614,137.815);

```

Im zweiten Abschnitt der Datei habe ich dann aus den Einzelpfaden die vier geschlossenen Kurvenzüge für die Länder erzeugt.

```

path Deutschland, Polen, TschRepu, Slowakei ;

Deutschland := dpts_map[1]--
    (reverse dpts_map[3])--
    (reverse dpts_map[2])--cycle ;

```

Hier ist nur beispielhaft der Pfad für Deutschland dargestellt, aber für die anderen Pfade gilt Entsprechendes. Stellenweise muss mit `reverse` der Richtungssinn eines Pfades angepasst werden, damit die Punkte der Pfade in der richtigen Reihenfolge vorliegen.

Darauf folgen die eigentlichen vier Grafiken. In jeder wird nur der Bereich eines Landes mit einer Farbe gefüllt. Das Ganze sieht dann ausschnittsweise so aus:

```

beginfig(1);
    fill Deutschland withcolor .7white ;
endfig;

```

```

...
beginfig(4);
  fill Slowakei withcolor .4white ;
endfig;
end

```

## Recycling

Nachdem nun der Aufbau der einzulesenden Datei klar ist, können wir uns an das versprochene Recycling machen. Geht man davon aus, dass die Datei mit `input` eingelesen wird, ergeben sich auf ersten Blick drei Probleme. Das sind `beginfig` und `endfig`, da wir uns in unserer eigentlichen Eingabe bereits in einem `beginfig-endfig`-Abschnitt befinden. Weiterhin das `end`, da wir keineswegs beim Einlesen den ganzen METAPOST-Lauf beenden wollen. Diese Probleme können durch geeignetes Umdefinieren der Befehle gelöst werden.

Betrachten wir die Befehle `beginfig` und `endfig` zunächst nicht, dann stellt sich die Datei als ein Strom Definitionen und Zeichenoperationen dar. Die Zeichenoperationen werden in METAPOST immer in der Variablen `currentpicture` ausgeführt. Beispielsweise sind folgende Operationen äquivalent:

```

fill Slowakei withcolor .4white ;

```

oder

```

addto currentpicture contour Slowakei withcolor .4white ;

```

Da Variablen in METAPOST, auch in einer Gruppe, immer global sind, würden die Zeichenoperationen direkt in die eigentliche Grafik zeichnen. Besser wäre es doch aber, wenn die einzulesende Grafik nachträglich noch skalierbar und frei positionierbar wäre. Das bedeutet, dass der Inhalt der Variablen `currentpicture` geschützt werden muss und das zu entwickelnde Makro die einzulesende Grafik in Form eines METAPOST-*picture* zurückgeben sollte. Dieses kann dann weiter transformiert werden.

Weiterhin sind auch nur die Zeichenoperationen der ausgewählten Grafik von Bedeutung. Deshalb muss durch geeignete Definition von `beginfig` und `endfig` eine Auswahl erfolgen. Das alles leistet das im Folgenden vorgestellte METAPOST-Makro.

Das Makro nimmt lokal eine Definition der Befehle `beginfig` und `endfig` vor, die an deren Originaldefinition angelehnt ist. Es berücksichtigt und schützt

somit auch einige globale Variablen. Das sind vor allem `x` und `y`, die gerne in METAPOST benutzt werden, da sie schon vordefiniert sind und die Einstellungen für den Zeichenstift `defaultpen`. Das heißt, die eingelesene Datei findet eine Umgebung vor, die möglichst dem normalen METAPOST-Lauf entspricht.

```
inner end ;
```

Dieses vorausgehende `inner end` ist notwendig, damit im Makro die Definition für `end` geändert werden kann.

```
vardef getextfigure(expr f, n) =
  save currentpicture, theExtFigure ;
  picture currentpicture; currentpicture := nullpicture ;
  numeric theExtFigure ; theExtFigure := n ;
```

Da das Makro das Ergebnis der Zeichenoperationen zurückgeben soll, wird es in Form eines `vardef`-Makros definiert. Diese Art der Definition erzeugt automatisch eine Gruppe. Dadurch sind die nachfolgend mit `save` geschützten Variablen lokal. Das Makro erwartet als Parameter den Namen der einzulesenen Datei `f` in Form eines METAPOST-*string* und die Nummer der gesuchten Grafik `n`, wie sie bei `beginfig` angegeben wurde.

Wie bereits oben beschrieben, wird die Variable `currentpicture` geschützt, damit die Zeichenoperationen der einzulesenen Datei nicht direkt in der eigentlichen Grafik wirken, sondern als *picture* zur weiteren Bearbeitung bereitstehen.

```
save extra_beginfig, extra_endfig;
string extra_beginfig, extra_endfig;
extra_beginfig = extra_endfig = "";

save beginfig, endfig, end ;
let end = relax ;
save endinputfig ;
```

Dieser Abschnitt schützt und initialisiert noch zwei weitere wichtige Variablen. Weiterhin werden die Makrodefinitionen von `beginfig`, `endfig` und `end` geschützt, da sie im Folgenden umdefiniert werden sollen. Das Makro `end` wird so definiert, dass es keine Wirkung hat.

Durch die unten folgende Konfiguration des Makros `endfig` ist normalerweise sichergestellt, dass der Befehl `end` beim Einlesen nie erreicht wird. Eine

Ausnahme ist jedoch, wenn der Nutzer eine Nummer für die Grafik angibt, die in der eingelesenen Datei gar nicht vorhanden ist. Somit könnte man `end` auch so definieren, dass es eine Warnung für diesen Fall ausgibt. Dies sollte aber leicht selbst implementierbar sein. Zusätzlich erfolgt noch der Schutz des Makros `endinputfig`. Dieses wird als Hilfsmakro nachfolgend benötigt.

```
def endfig =
  scantokens extra_endfig;
  endgroup;
  currentpicture := nullpicture ;
enddef;

def endinputfig =
  scantokens extra_endfig;
  endgroup;
  endinput;
enddef;
```

Die Definition von `endfig` erfolgt analog zum Original. Jedoch werden hier statt des Aufrufs der Ausgaberoutine `shipit` die erfolgten Zeichenoperationen wieder verworfen, indem die Variable `currentpicture` zurückgesetzt wird. Das bedeutet, dass in dieser Grundkonfiguration alle Grafikoperationen ignoriert werden. Das Hilfsmakro `endinputfig` ist ähnlich definiert. Jedoch führt es nicht zum Rücksetzen der Variablen `currentpicture`, sondern durch Aufruf von `endinput` zum Abbruch eines Einleseprozesses. Wieso das nützlich ist, wird in der folgenden Definition klar:

```
def beginfig (expr i) =
  if (i=theExtFigure): let endfig = endinputfig ; fi;
  begingroup;
  clearxy; clearpen;
  pickup defaultpen;
  drawoptions();
  scantokens extra_beginfig;
enddef;
```

Das Makro `beginfig` verhält sich wie das Original. Es werden die Variablen `x` und `y` durch `clearxy` und der Zeichenstift `defaultpen` initialisiert. Entscheidender ist jedoch die erste Zeile der Definition. Hier wird verglichen, ob die Nummer der Grafik mit der Nummer der gesuchten Grafik übereinstimmt. Wenn dem so ist, dann wird `endfig` auf die Definition von `endinputfig`

gesetzt. Das bedeutet, dass in der Grundkonfiguration `endfig` alle Zeichenoperationen verwirft. Stammen jedoch die Zeichenoperationen von der gesuchten Grafik, dann wird am Ende der Grafik der Einlesevorgang abgebrochen. Damit ist die gesuchte Grafik in der lokalen Variablen `currentpicture` gespeichert.

```
scantokens ("input " & f ) ;
currentpicture
enddef;
```

Letztendlich liest das Makro die externe Datei ein. Der etwas kompliziert wirkende Einleseaufruf `scantokens ("input " & f )` ist notwendig, da METAPOST ein `input` in Makros eigentlich nicht zulässt. Der Rückgabewert ist der Inhalt der Variablen `currentpicture`. Wurde beim Aufruf eine gültige Grafiknummer angegeben, dann enthält der Rückgabewert das Ergebnis der Zeichenoperationen in dieser Grafik. Im Fall einer nicht existierenden Grafik mit der angegebenen Nummer ist der Rückgabewert `nullpicture`, also ein leeres METAPOST-*picture*.

Beispielsweise kann man so die Länder Polen und Deutschland aus der Datei `dpts.mp` auslesen und in einer anderen Datei wiederverwenden.

```
beginfig(1);
picture pic ;
pic := gettextfigure("dpts.mp",2) ;% Polen
addto pic also gettextfigure("dpts.mp",1) ;% Deutschland
draw pic scaled 0.7 ;
endfig;
```

Somit beinhaltet vor dem `draw` die Variable `pic` zwei gefüllte Pfade mit den Umrissen von Polen und Deutschland. Normalerweise wird man die Grafik noch weiter ergänzen wollen. Ein Beispiel zeigt Abbildung 2. Hier wurden noch Daten zu den Ländern in Form von Tortendiagrammen eingefügt. Die Diagramme wurden mit Hilfe des Pakets `piechartMP` [6] erstellt.

## Probleme

Das im vorherigen Abschnitt dargestellte Macro schützt nur die bekannten problematischen Definitionen. Dazu zählen in erster Linie natürlich die Makros `beginfig` und `endfig` aber auch andere Variablen wie `x` und `y`.

Da in METAPOST jedoch alle Variablen global sind, es sei denn, sie sind durch `save` innerhalb einer Gruppe geschützt, kann es zu Problemen kommen, wenn

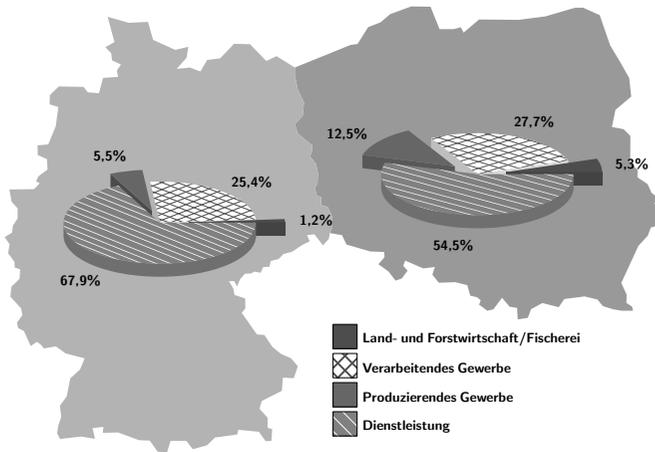


Abbildung 2: Die eingelesenen Grafiken der Länder Deutschland und Polen, ergänzt mit Daten über die prozentuale Zusammensetzung des Bruttoinlandprodukts 2001. (Quelle: Statistisches Bundesamt)

eine eingelesene Datei die gleichen Variablen wie die einlesende Datei benutzt. Da das Makro über die verwendeten Variablen keine Informationen hat, bleibt es dem Nutzer überlassen, dafür zu sorgen, dass Variablenwerte aus der einlesenden Datei sich nicht negativ auf das Verhalten der eingelesenen Datei auswirken und umgekehrt. Die folgende Definition schützt beispielsweise die Variable `var`:

```
beginfig(1);
picture pic ; numeric var ; var := 0.7 ;

begingroup;
save var;
pic := gettextfigure("dpts.mp",2) ;
addto pic also gettextfigure("dpts.mp",1) ;
endgroup;

draw pic scaled var ;
endfig;
```

Aber ich brauche noch...

So beginnt ein Satz, bei dem man schon jetzt weiß: Es macht Arbeit! Richtig verloren fühlt man sich dann aber, wenn der Satz mit „...eine Abbildung für die Tschechoslowakei“ weitergeht. Natürlich hat man seine Karten auf den aktuellen politischen Stand, auch wenn einige Kräfte ständig am wirken sind [1], um Veränderungen herbeizuführen, die manchmal eine Aktualisierung erforderlich machen. Nun soll es jedoch zurück in die Geschichte gehen. Zurück ins Jahr 1991, denn da bildeten die heutige Tschechische Republik und die Slowakei noch ein Land – die Tschechoslowakei.

In der externen Datei `dpts.mp` sind die beiden Grafiken für diese Länder in Grafik 3 und 4 bekanntlich vorhanden. Problematisch ist nur, dass sie mit unterschiedlichen Farben gefüllt sind. Die Füllfarbe der Tschechei ist `0.5white`, die der Slowakei ist `0.4white`. Das bedeutet, dass zumindest für eine der Grafiken eine Farbanpassung erfolgen muss, damit sie zusammen den Eindruck einer einheitlich gefüllten Fläche für die Tschechoslowakei ergeben.

Für den konkreten Fall dieser einfachen externen Datei gibt es auch noch eine unkomplizierte Lösung, die hier aber auch genannt werden soll, da sie nicht sofort offensichtlich ist.

```
beginfig(1);
picture CSR ;% Tschechoslowakei

CSR := gettextfigure("dpts.mp",3) ;
addto CSR also gettextfigure("dpts.mp",4) withcolor .5white;

draw CSR ;
endfig;
```

Mit dem `withcolor .5white` werden *alle* Grafikelemente in dieser eingelesenen Grafik der Slowakei entsprechend eingefärbt. Da die Grafik nur aus einem gefüllten Pfad besteht, ist somit das Ziel schon erreicht, dass dieser dann die gleiche Farbe wie die Fläche der Tschechischen Republik hat. Bei komplizierteren Grafiken wünscht man sich aber sicher ein selektiveres Verhalten.

Die Funktionen, die das ermöglichen, gehören in meinen Augen zu den stärksten Features von METAPOST. Leider sind sie nicht in der normalen Anleitung zu METAPOST [4] dokumentiert, sondern sie finden sich im Anhang der Dokumentation zum Paket `graph` [3].

Zentraler Ausgangspunkt ist das Makro `within`. Man kann es sich so vorstellen, dass dieses Makro eine Liste aller Grafikelemente eines `METAPOST-picture` liefert. Grafikelemente sind gezeichnete Linien, gefüllte Flächen, Textelemente aber auch Pfade zum Beschneiden (`clipping`) und für die Bounding Box (`setbounds` Pfade). Die Liste kann beispielsweise in einer Schleife durchlaufen werden. So durchläuft folgende Schleife alle Grafikelemente der aktuellen Grafik (`Variable currentpicture`) und stellt sie in Form der Schleifenvariablen `i` zur Verfügung:

```
for i within currentpicture:
  ...
endfor;
```

Um nun damit Grafikelemente modifizieren zu können, denn nichts anderes stellt die Änderung einer Farbe dar, muss es natürlich möglich sein, Informationen über das Grafikelement zu erhalten. Der Typ des Grafikelements kann mit den Makros `stroked`, `filled`, `textual`, `clipped` und `bounded` ermittelt werden. Die Makros geben `true` oder `false`, je nachdem ob die Eigenschaft auf das Grafikelement zutrifft. Die Makros `clipped` und `bounded` betreffen Pfade, die Grafikelemente beschneiden oder die Bounding Box der Grafik ändern. Diese Elemente bedürfen somit einer Sonderbehandlung. Die verbleibenden drei Makros testen auf gezeichnete Linien (`stroked`), auf gefüllte Flächen (`filled`) und auf Textelemente (`textual`).

Jedoch nützt uns das alles nichts, wenn es nicht möglich ist, weitere Informationen, wie beispielsweise die Farbe des Grafikelements, zu erhalten. Dafür wurden weitere Makros bereitgestellt. Beispielsweise gibt `pathpart` den Pfad für Linien oder gefüllte Flächen. Die Farbwerte eines Elements können mit `redpart`, `greenpart` und `bluepart` ermittelt werden. Es gibt noch einige weitere dieser Makros, aber auf diese einzugehen, verlangt mehr, als in diesem Artikel möglich ist. Trotzdem werden sie im Folgenden benutzt, um eine möglichst allgemein verwendbare Basisdefinition zu zeigen. Da die Makros aber sinnvolle Namen tragen, sollte es einfach möglich sein, ihre Funktion zu erkennen.

Um zur ursprünglichen Problematik zurückzukommen, geht es im folgenden um ein Makro `uniformfilled`, das die Farbe aller gefüllten Grafikelemente eines `METAPOST-picture` mit einer anderen Farbe ersetzt. Die Anwendung erfolgt beispielsweise für das oben beschriebene Problem so:

```
beginfig(1);
picture CSR ;% Tschechoslowakei
```



Abbildung 3: Originalgrafik und Ergebnis nach Anwenden von `uniformfilled` auf die Originalgrafik

```
CSR := gettextfigure("dpts.mp",3) ;% Tschechische Republik
addto CSR also gettextfigure("dpts.mp",4) ;% Slowakei

draw CSR uniformfilled 0.6white ;
endfig;
```

Da das bisherige Beispiel nur aus gefüllten Flächen besteht und somit die Selektivität auf gefüllte Flächen schlecht illustrierbar ist, möchte ich hier zu einer anderen Grafik wechseln. Sie ist in Abbildung 3 links dargestellt. Das Ergebnis nach Anwenden des Makros zeigt die rechte Grafik in der selben Abbildung.

Die Grafik besteht aus vier quadratischen Flächen, die wechselnd mit `0.8white` und `0.5white` gefüllt sind. Diese vier Flächen wurden im Quadrat angeordnet und auf den ebenfalls dargestellten Kreis beschnitten (clipping). Weiterhin enthält die Grafik eine horizontale und vertikale Linie in Farbe `white` und einen Schriftzug. Nachdem das Makro auf die Grafik angewendet wurde, sind alle gefüllten Flächen einheitlich mit `0.6white` gezeichnet. Die Farben der anderen nichtgefüllten Grafikelemente behalten, wie in der Abbildung 3 rechts zu sehen, ihren ursprünglichen Farbzustand.

Das Makro `uniformfilled` erwartet ein `METAPOST-picture` für Variable `p` und eine Farbe für Variable `c`. Das Makro arbeitet vollständig in einer Gruppe, da noch drei lokale Hilfsvariablen benötigt werden.

```
primarydef p uniformfilled c =
  begingroup
  save t, f, s ; picture t, s ; color f ;
  t := nullpicture ;
```

In die Variable `t` vom Typ *picture* werden alle Grafikelemente der Grafik `p` übernommen. Dabei werden die Grafikelemente unter Umständen modifiziert. In diesem Fall betrifft es hier die Füllfarbe von Flächen. Die Farbe `f` enthält die ursprüngliche Farbe des Grafikelements. Sie wurde nur aus Gründen der Übersichtlichkeit eingeführt, da es kein Makro `colorpart` gibt, sondern man nur mit `redpart`, `greenpart` und `bluepart` die Farbe eines Grafikelements bestimmen kann.

Anschließend werden alle Grafikelemente der Variablen `p` in einer Schleife durchlaufen. Das einzelne Grafikelement ist als Variable `i` verfügbar.

```
for i within p :
  f := (redpart i, greenpart i, bluepart i) ;
  if bounded i :
    s := i uniformfilled c ;
    setbounds s to pathpart i ;
    addto t also s ;
  elseif clipped i :
    s := i uniformfilled c ;
    clip s to pathpart i ;
    addto t also s ;
```

Im ersten Abschnitt erfolgt die Sonderbehandlung für Grafikelemente, die beschnitten sind (`clipped`) oder die Bounding Box modifizieren (`bounded`). Diese Grafikelemente können wiederum weitere Grafikelemente enthalten, auf die sich die Operation, also beispielsweise das Beschneiden, bezieht. Aus diesem Grund wird das Makro `uniformfilled` nochmals rekursiv auf das Grafikelement `i` angewendet.

```
elseif stroked i :
  addto t doublepath pathpart i dashed dashpart i
    withpen penpart i withcolor f ;
elseif filled i :
  addto t contour pathpart i withcolor c ;
```

Im nächsten Abschnitt werden Linienelemente (`stroked`) und gefüllte Flächen (`filled`) behandelt. Die Linienelemente werden in ihrer ursprünglichen Erscheinung übernommen. Hier ist schön zu sehen, wie die verschiedenen Attribute mittels `pathpart`, `penpart` etc. ermittelt werden können. Die Farbe für Linienelemente ist die ursprüngliche Farbe `f`, die am Anfang der Schleife ermittelt wurde. Im Gegensatz dazu werden gefüllte Flächen (`filled`) mit der Farbe gefüllt, die dem Makro als Variable `c` übergeben wurde. Dadurch wird die selektive Farbanpassung erreicht.

```
    else:
        addto t also i ;
    fi ;
endfor;
t
endgroup
enddef;
```

Im letzten Abschnitt der Schleife werden alle weiteren Grafikelemente behandelt. Darunter fällt im Beispiel auch das Textelement. Man könnte zusätzlich noch auf `textual` prüfen und dann mittels `textpart` und `fontpart` das Textelement hinzufügen. Jedoch sind dabei die Transformationen des Textelements noch nicht berücksichtigt. Da die Behandlung von Textelementen nicht das Ziel war, habe ich diese einfache Lösung gewählt.

Das Makro gibt letztendlich den Inhalt der Variablen `t` zurück, die alle Grafikelemente enthält.

## Zusammenfassung

METAPOST bietet mit Unterstützung einfacher Makros die Möglichkeit, auf bereits vorhandene Grafiken zurückzugreifen. Damit lassen sich beispielsweise Grafikbibliotheken aufbauen oder komplexe Grafiken mit geringem Aufwand wiederverwenden. Nachträglich können diese Grafiken beispielsweise noch an das Farbschema des neuen Projekts angepasst werden, um ein einheitliches Erscheinungsbild zu erreichen. Die Möglichkeiten der nachträglichen Anpassung sind vielfältig und natürlich nicht nur auf Farben beschränkt. Eine Behandlung würde aber den sinnvollen Rahmen dieses Artikels sprengen. Somit bleibt dies eigenen Experimenten vorbehalten, für die hier, wie ich hoffe, ein paar Anregungen gegeben wurden.

## Literatur

- [1] Michel Chossudovsky: *Global brutal*; Zweitausendeins; Frankfurt am Main; 2002.
- [2] Wolfgang Glunz: *pstoedit*; <http://www.pstoedit.com>.
- [3] John D. Hobby: *Drawing Graphs with METAPOST*; AT&T Bell Laboratories; Murray Hill, New Jersey; CTAN:systems/msdos/metapost/doc/mpgraph.ps.

- [4] John D. Hobby: *A User's Manual for METAPOST*; AT&T Bell Laboratories; Murray Hill, New Jersey; April 1992; CTAN:systems/msdos/metapost/doc/mpman.ps.
- [5] Lauris Kaplinski: *SodiPodi – A Vector Drawing Program*; <http://sodipodi.sourceforge.net>.
- [6] Jens-Uwe Morawski: *piechartMP – drawing pie-charts with METAPOST*; CTAN:graphics/metapost/macros/piechartmp/.
- [7] Gerd Neugebauer: *Illustrationen für L<sup>A</sup>T<sub>E</sub>X erstellen; Die T<sub>E</sub>Xnische Komödie*; 2/2002, S. 46–48; 2002.
- [8] Michael Ritz: *Landkarten und Stadtplan Index*; <http://www.kartenindex.de>.

# Von fremden Bühnen

## Der `\year=2003` T<sub>E</sub>X Kalender

T<sub>E</sub>X Merchandising Project, Martin Schröder

Das T<sub>E</sub>X Merchandising Project präsentiert den `\year=2003` T<sub>E</sub>X Kalender:

Die Titelseite



Bild für Juli



Kalendarium für Juli

Juli						
Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Sonntag	
22	23	24	25	26	27	28
29	30	31				

Juli						
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
27	28	29	30	31		

August						
Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Sonntag	
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

- o Format ISO A4, 29 Seiten
- o Jeder Monat mit einem Bild von Duane Bibby, acht davon aus den Büchern von Donald Knuth. Titelbild von Duane Bibby exklusiv für diesen Kalender
- o Kalendarium dreisprachig in deutsch/englisch/französisch mit T<sub>E</sub>X-Gedenktagen und jeweils drei Monaten auf einen Blick
- o Kurze Einführung in T<sub>E</sub>X und METAFONT
- o Spiralbindung zum Umschlagen mit Bügel zum Aufhängen
- o Preis: 12,95 €

- ISBN: 3-936427-20-8
- Alle Erlöse kommen dem T<sub>E</sub>X Merchandising Fund zugute, der von DANTE e.V. verwaltet wird

*Das ideale Weihnachtsgeschenk für alle T<sub>E</sub>X-Nutzer und -Nochnichtnutzer!*

Bezugsquelle:

- Lehmanns Fachbuchhandlung,  
Abt. Versand,  
Hardenbergstraße 11,  
10623 Berlin,  
Tel. 0800 / 2 66 26 65,  
Fax (+49) 030 / 61 79 11 33,  
E-Mail [bestellung@lehmanns.de](mailto:bestellung@lehmanns.de),  
<http://www.lob.de>

# Spielplan

---

## Termine

- 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<sub>E</sub>X 2003  
14<sup>th</sup> Annual Meeting of the European T<sub>E</sub>X Users' Group  
Bretagne, Frankreich  
Kontakt: Secrétariat de la conférence EuroT<sub>E</sub>X 2003
- 20.–26. 7. 2003** TUG 2003 Waikaloa Beach Resort, Big Island, Hawaii  
<http://www.tug.org/tug2003/>  
Kontakt: Wendy McKay
- 8.–9. 9. 2003** 29. Mitgliederversammlung von DANTE e.V.  
Schloß Rauischholzhausen (bei Marburg),  
Universität Gießen, Gießen  
Kontakt: DANTE e.V.

## Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T<sub>E</sub>X-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](mailto: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](mailto: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](mailto: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](mailto:Stephan.Lehmke@cs.uni-dortmund.de)  
Café Durchblick  
Universität Dortmund, Campus Nord  
Zweiter Mittwoch im Monat, 20.00 Uhr

### Dresden

Julia Brückke  
[jule.brueck@web.de](mailto:jule.brueck@web.de)  
Medien- und Kulturzentrum Pentacon,  
Schandauer Str. 64  
Erster Mittwoch im Monat, 19.00 Uhr

### Erlangen

Walter Schmidt, Peter Seitz  
[was@VR-Web.de](mailto:was@VR-Web.de),  
Gaststätte „Erlanger Gärtla“  
Marquardsenstraße 1  
Dritter Dienstag im Monat, 19.00 Uhr

### Freiburg

Heiko Oberdiek  
Tel.: 07 61/4 34 05  
[oberdiek@uni-freiburg.de](mailto:oberdiek@uni-freiburg.de)  
Gaststätte „Aquila“  
Sautierstr. 19  
Dritter Donnerstag im Monat, 19.30 Uhr

### Hamburg

Volker Hüttenrauch  
[volker\\_huettenrauch@hh.maus.de](mailto: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](mailto: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](mailto: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](mailto: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@fieee.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<sub>E</sub>X e.V.  
Postfach 10 18 40  
69008 Heidelberg

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10<sup>00</sup>–12<sup>00</sup> Uhr)  
Fax: 0 62 21/16 79 06  
E-Mail: [dante@dante.de](mailto: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	<a href="mailto:president@dante.de">president@dante.de</a>
Vizepräsident:	Klaus Höppner	<a href="mailto:vice-president@dante.de">vice-president@dante.de</a>
Schatzmeister:	Tobias Sterzl	<a href="mailto:treasurer@dante.de">treasurer@dante.de</a>
Schriftführer:	Günter Partosch	<a href="mailto:secretary@dante.de">secretary@dante.de</a>
Beisitzer:	Thomas Koch	
	Bernd Raichle	<a href="mailto:advisor@dante.de">advisor@dante.de</a>

## Server

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

## Autoren/Organisatoren

<b>TEX Merchandising Project</b> p. A. Martin Schröder Crüsemannallee 3 28213 Bremen tm@tm.oneiros.de <a href="http://www.tm.oneiros.de">http://www.tm.oneiros.de</a>	[63]	<b>Wendy McKay</b> wgm@cads.caltech.edu	[65]
<b>Secrétariat de la conférence EuroTEX 2003</b> Département Informatique ENST Bretagne France Yannis.Haralambous@enst-bretagne.fr	[65]	<b>Jens-Uwe Morawski</b> Gryphiustraße 9A 10245 Berlin	[48]
<b>Klaus Höppner</b> siehe Seite 68	[4]	<b>Gerd Neugebauer</b> Im Lerchelsbühl 5 64521 Groß-Gerau gene@gerd-neugebauer.de	[3]
<b>David Kastrup</b> Kriemhildstr. 15 44793 Bochum David.Kastrup@t-online.de	[10]	<b>Rolf Niepraschk</b> Persiusstr. 12 10245 Berlin niepraschk@ptb.de	[25]
<b>Markus Kohm</b> Fichtenstraße 63 68535 Edingen-Neckarhausen kohm@gmx.de	[28]	<b>Günter Partosch</b> Schriftführer von DANTE e.V. Guenter.Partosch@hrz.uni-giessen.de	[5]

# Die T<sub>E</sub>Xnische Komödie

---

14. Jahrgang Heft 4/2002 Dezember 2002

## Impressum

## Editorial

## Hinter der Bühne

- 4 Grußwort
- 6 Beschlüsse der 27. Mitgliederversammlung von DANTE e.V. am 5. Oktober 2002 in Augsburg

## Bretter, die die Welt bedeuten

- 10 L<sup>A</sup>T<sub>E</sub>X und WYSIWYG? preview-latex unter Emacs und andere Ansätze
- 27 Tipps und Tricks: Immer im Rahmen bleiben
- 28 Satzspiegelkonstruktionen im Vergleich
- 48 Aus alt mach neu – METAPOST Recycling

## Von fremden Bühnen

- 63 Der \year=2003 T<sub>E</sub>X Kalender

## Spielplan

- 65 Termine
- 66 Stammtische

## Adressen

- 69 Autoren/Organisatoren