

# Die T<sub>E</sub>Xnische Komödie

---

dante

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

21. Jahrgang Heft 4/2009 Dezember 2009

4/2009

# Impressum

---

»Die  $\TeX$ nische 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- $\LaTeX$ -Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und per E-Mail oder Datenträger (CD) an untenstehende Adresse der Redaktion geschickt werden. Sind spezielle Makros,  $\LaTeX$ -Pakete oder Schriften dafür nötig, so müssen auch diese komplett mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit `pdfTeX 3.1415926-1.40.10-2.2` erstellt. Als Standard-Schriften kamen die Type-1-Fonts Latin-Modern und LuxiMono zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung  $\TeX$  e.V.

Postfach 10 18 40

69008 Heidelberg

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

[dtkred@dante.de](mailto:dtkred@dante.de) (Redaktion)

Druck: Konrad Tritsch Print und digitale Medien GmbH

Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohestadt

Redaktion: Herbert Voß (verantwortlicher Redakteur)

Mitarbeit : Gert Ingold                      Rolf Niepraschk                      Christine Römer

Volker RW Schaa

Redaktionsschluss für Heft 1/2010: 15. Januar 2010

ISSN 1434-5897

# Editorial

---

Liebe Leserinnen und Leser,

Wunder geschehen ja eigentlich nicht in regelmäßigen Abständen, jedenfalls wären es dann definitionsgemäß keine mehr. So bleibt das Problem, wie man es bezeichnen soll, was Sie in den Händen halten. Dass bekanntermaßen nach einer Ausgabe der Zeitschrift »Die T<sub>E</sub>Xnische Komödie« natürlich immer auch vor einer Ausgabe ist, wird keinen erstaunen. Wenn aber nach einer neuen »Komödie« der »Eingabespeicher« der eingereichten Manuskripte immer vollständig geleert ist, so bleibt es ein Wunder, dass bis zur nächsten Ausgabe dieser »Speicher« so aufgefüllt wird, dass es wieder für ein neues Heft in gewohntem Umfang reicht. Der Dank gebührt natürlich den Autorinnen und Autoren, denn ohne sie könnte es tatsächlich zu einem echten Wunder werden.

Dass T<sub>E</sub>X, als reines Textsatzsystem konzipiert, durch derartig viele verschiedene Pakete zur Grafikausgabe unterstützt wird, ist ebenfalls ein kleines Wunder, zumal die Qualität mit der vom Textsatz gewohnten durchaus konkurrieren kann. In diesem Heft finden Sie wieder eine Variante beschrieben. Weiterhin lesen Sie den Bericht von der letzten Herbsttagung in Esslingen und dem Linuxtag in Kiel, auf dem DANTE e.V. erstmalig vertreten war. Außerdem finden Sie Texte, die eher den praktischen Bereich von L<sup>A</sup>T<sub>E</sub>X beleuchten und natürlich, wie gewohnt, die Liste der neuen Pakete. Der Umfang dieser Liste ist das letzte Wunder von dem man berichten kann; denn nach über 30 Jahren T<sub>E</sub>X und 20 Jahren L<sup>A</sup>T<sub>E</sub>X erscheint die innovative Kraft der T<sub>E</sub>X-Gemeinde ungebrochen.

Ich wünsche Ihnen wie immer viel Spaß beim Lesen und verbleibe  
mit T<sub>E</sub>Xnischen Grüßen,  
Ihr Herbert Voß

# Hinter der Bühne

---

Vereinsinternes

## Grußwort

Liebe Mitglieder,

was lange währt, wird endlich gut, sagt der Volksmund. So halten Sie nun, hoffentlich vor Weihnachten oder spätestens ein paar Tage danach, die letzte Ausgabe des Jahres 2009 von »Die T<sub>E</sub>Xnische Komödie« in den Händen.

Dass es so lange dauerte, lag daran, dass die Produktion der diesjährigen DVD »T<sub>E</sub>XCollection« unter keinem guten Stern stand. Während der Mitgliederversammlung in Esslingen klangen wir noch sehr optimistisch bezüglich der Veröffentlichung von T<sub>E</sub>X Live. Leider erwies sich diese Hoffnung als trügerisch, denn kurz danach ging begann eine lange Diskussion innerhalb des Teams von T<sub>E</sub>X Live. Der Knackpunkt war das Feature, das auf den Namen `\write18` hört und sich dann schnell als Büchse der Pandora erwies. Es dient dazu, während der Ausführung des T<sub>E</sub>X-Compilers externe Programme aufzurufen. Diese Eigenschaft war in der Vergangenheit aus Sicherheitsgründen standardmäßig deaktiviert. Nun ist aber beim Übersetzen von Dokumenten mit pdfL<sup>A</sup>T<sub>E</sub>X sehr hinderlich, dass genau das Grafikformat EPS, das auf dem herkömmlichen Weg mit DVI als Ausgabeformat und Konvertierung nach PostScript mit `dvips` in der Regel verwendet wurde, von pdfL<sup>A</sup>T<sub>E</sub>X nicht unterstützt wird. Daher entstand die Idee, mit Hilfe von `\write18` von pdfL<sup>A</sup>T<sub>E</sub>X aus automatisch EPS durch Aufruf von `epstopdf` in das unterstützte Grafikformat PDF umzuwandeln. Hierfür sollte `\write18` also nun aktiviert werden, allerdings nur für eine eng begrenzte Menge von vertrauenswürdigen Programmen. Dies sollte verhindern, dass durch L<sup>A</sup>T<sub>E</sub>X-Dokumente, die allgemein als ungefährlich gelten, auf einmal über den Aufruf externer Programme mutwillig Schaden angerichtet werden kann. Leider erwies sich diese scheinbare Sicherheit durch Beschränkung auf bestimmte Programme mehr als Wunschdenken denn als wirklicher Schutz, so gelang es einigen Personen, Angriffsszenarien aufzuzeigen. Letztlich wurde daher kurzfristig entschieden, das `\write18` wieder standardmäßig abzuschalten und auf die neue, bereits öffentlich angekündigte Errungenschaft der automatischen Konvertierung von EPS-Grafiken nach PDF

zu verzichten (und soweit in der Dokumentation von T<sub>E</sub>X Live stellenweise noch auf dieses Feature Bezug genommen wird, bitten wir dies nachzusehen).

Für uns hatte dies leider die unangenehme Folge, dass die endgültige Version von T<sub>E</sub>X Live 2009 erst im November erschien. Dann hieß es wieder warten, denn MacT<sub>E</sub>X als ein immer wichtigerer Teil der DVD enthält T<sub>E</sub>X Live, so dass die Arbeit am T<sub>E</sub>X für die Apple-Fans dann erst richtig losging. Parallel stellte dann Manfred Lotz den Abzug von CTAN zusammen, was ihn wieder vor die Aufgabe stellte, ein immer größer werdendes Software-Archiv in immer weniger Platz auf der DVD, der von T<sub>E</sub>X Live und MacT<sub>E</sub>X übrig gelassen wird, unterzubringen. Fast in letzter Minute wurde dann noch beim Testen ein Fehler in MacT<sub>E</sub>X gefunden, dessen Korrektur wieder einige Tage in Kauf nahm. Den allerletzten Schreck gab es dann, als die DVDs sich schon per Express auf dem Weg zum Presswerk befanden, und es auf einmal so schien, als wäre immer noch ein Fehler in MacT<sub>E</sub>X. Dies erwies sich zum Glück als falscher Alarm, so dass die Produktion dann doch losging.

Sie sehen, die Produktion einer solchen DVD ist komplex, und leider kostet sie auch Nerven.

Ein positiver Nebeneffekt der Verzögerung ist allerdings, dass in diese Ausgabe von »Die T<sub>E</sub>Xnische Komödie« gerade noch ein Artikel von Idris Samawi Hamid hineinkam. Er beschreibt darin das Projekt Oriental T<sub>E</sub>X, das einerseits von der Colorado State University und andererseits von verschiedenen T<sub>E</sub>X-Gruppen, darunter auch DANTE e.V., gefördert wurde. Erscheinen uns Europäern die Feinheiten der arabischen Typografie auch sehr fremd, so war dieses Projekt Anlass und Inspiration für viele Ergänzungen in luaT<sub>E</sub>X, um im Satz Dinge möglich zu machen, die mit dem konventionellen T<sub>E</sub>X-Compiler so nicht oder nur schwer realisierbar sind. Idris Samawi Hamid hat einen sehr interessanten Vortrag während der Konferenz TUG 2009 in den USA gehalten, wir hoffen, dass Sie seinen Artikel ebenso interessant finden. Wenn es seine Zeit erlaubt, will Idris Samawi Hamid auch auf der nächsten Frühjahrstagung von DANTE e.V. einen Vortrag halten.

Die kommende Frühjahrstagung DANTE 2010 findet Anfang März an der Technischen Universität Dortmund statt, an der wir somit nach 1999 zum zweiten Mal zu Gast sein werden. Die Einladung zur Tagung und Mitgliederversammlung mit dem »Call for Papers« finden Sie ebenfalls in dieser Zeitschrift. Wie angekündigt findet dort regulär die Neuwahl des Vorstands statt, natürlich begleitet von den üblichen Formalien wie dem Bericht der Kassenprüfer und der Entlastung des Vorstands.

Leider sind in Anbetracht der Dinge, die dringend getan werden mussten, andere Dinge liegen geblieben. Beim Thema »T<sub>E</sub>Xnische Kontakte«, dem

schon lange nicht mehr erschienenen gedruckten Mitgliederverzeichnis, ist mittlerweile aber das sprichwörtliche Licht am Ende des Tunnels sichtbar (mit der Hoffnung, dass es nicht der entgegenkommende Zug ist).

Seit dem Größwort in der letzten Ausgabe fand in den Niederlanden die diesjährige EuroT<sub>E</sub>X statt. Es war eine sehr angenehme Tagung, deren Veranstaltungsort eine Militärakademie in Delft war, wo sich dann lauter zivil gekleidete »T<sub>E</sub>Xies« zwischen Uniformträgern tummelten, die nebenan über Themen wie die Sicherheitslage in Afrika tagten.

Die bedeutenden Tagungen des kommenden Jahres werden die TUG 2010 im Juni in San Francisco sein (aus Anlass des 32. Geburtstages von T<sub>E</sub>X unter Teilnahme von Donald E. Knuth und weiteren wichtigen Personen der Frühzeit von T<sub>E</sub>X) sowie die EuroT<sub>E</sub>X 2010 Ende August in Pisa sein. Weiterhin werden wir als DANTE e.V. wieder mit Ständen auf den Chemnitzer Linuxtagen im März und dem Linuxtag in Berlin im Juni vertreten sein.

Zum Abschluss bleibt uns nur noch, Ihnen frohe Weihnachten und einen guten Rutsch ins neue Jahr zu wünschen (zumindest nachträglich, wenn dieses Exemplar doch nicht mehr rechtzeitig bei Ihnen ankam).

Mit freundlichem Größ,

Klaus Höppner	Volker RW Schaa
Vorsitzender	Stellvertretender Vorsitzender

## DANTE 2010 – Einladung zur Mitgliederversammlung und »Call for Papers«

Klaus Höppner, Martin Schröder

Liebe Mitglieder von DANTE e.V.,

die T<sub>E</sub>X-Tagung DANTE 2010 findet vom 3. bis zum 5. März 2010 an der Technischen Universität Dortmund statt. Veranstalter sind DANTE e.V. und die Fakultät für Informatik der TU Dortmund.

Am Mittwoch sind Tutorien geplant, Donnerstag und Freitag sind für Vorträge und die 42. Mitgliederversammlung von DANTE e.V. vorgesehen. Die Tagesordnung der Mitgliederversammlung am Donnerstag, den 4. 3. 2010 um 9.00 Uhr in der

Technischen Universität Dortmund  
Fakultät für Informatik  
Otto-Hahn-Straße 14  
44221 Dortmund

lautet:

1. Begrüßung und Tagesordnung
2. Bericht des Vorstands
3. Finanzbericht
4. Bericht der Kassenprüfer
5. Entlastung des Vorstands
6. Wahl des Vorstands
7. Wahl von Kassenprüfern
8. Verschiedenes

Ihre Stimmunterlagen erhalten Sie direkt vor Ort, um vorherige Anmeldung wird gebeten. Eine Übertragung des Stimmrechts ist im Rahmen des § 13 (4) der Vereinssatzung möglich. Wie üblich sind auch Nichtmitglieder als Gäste willkommen.

Falls Sie ein Tutorium oder einen Vortrag anbieten wollen, werden Sie gebeten, dies mit dem entsprechenden Formular auf der WWW-Seite der Tagung oder per E-Mail an [dante2010@dante.de](mailto:dante2010@dante.de) möglichst bald anzumelden. Fügen Sie

hierfür bitte eine Kurzzusammenfassung (Abstract) als Text- oder L<sup>A</sup>T<sub>E</sub>X-Datei bei.

Unter <http://www.dante.de/events/dante2010.html> finden Sie die Homepage der Tagung mit allen weiteren Informationen.

Mit Fragen, Wünschen und Anregungen wenden Sie sich bitte an

DANTE e.V.

Stichwort: DANTE 2010

Postfach 10 18 40

69008 Heidelberg

E-Mail: [dante2010@dante.de](mailto:dante2010@dante.de)

Mit freundlichen Grüßen,

Klaus Höppner (DANTE e.V.)

Martin Schröder (QuinScape GmbH)

## Beschlüsse der 41. Mitgliederversammlung von DANTE e.V. am 12. September 2009 in Esslingen

Manfred Lotz

Zeit: 12. September 2009, 9:15 Uhr – 10:50 Uhr

Ort: Hochschule Esslingen  
Standort ES Stadtmitte  
Gebäude 7 (Eingang Mühlstr. 3), 1. Stock  
Kanalstr. 33  
73728 Esslingen

Teilnehmer: 25 (anhand der ausgegebenen Stimmkarten)

Leitung: Klaus Höppner (Vorsitzender von DANTE e.V.)

Protokollant: Manfred Lotz (Schriftführer von DANTE e.V.)

Die Mitgliederversammlung wurde satzungsgemäß eingeladen und ist beschlußfähig.

### TOP 1: Begrüßung, Tagesordnung und Vorstellung des Vorstands

#### TOP 1.1: Begrüßung und Tagesordnung

Klaus Höppner begrüßt die Teilnehmer der 41. Mitgliederversammlung von DANTE e.V. in Esslingen und stellt die Tagesordnung vor:

1. Begrüßung, Tagesordnung und Vorstellung des Vorstands
  - Begrüßung und Tagesordnung
  - Vorstellung des Vorstands
2. Bericht des Vorstands
3. Verschiedenes

Die Tagesordnung wird ohne Einspruch akzeptiert.

#### TOP 1.2: Vorstellung des Vorstands

Alle derzeitigen Vorstandsmitglieder sind anwesend und werden von Klaus Höppner vorgestellt: Klaus Höppner (Vorsitzender), Volker RW Schaa (stellvertretender Vorsitzender), Tobias Sterzl (Schatzmeister), Manfred Lotz (Schriftführer), Günter Partosch (Beisitzer), Herbert Voß (Beisitzer) und Bernd Raichle (Beisitzer).

Der Verein unterhält in Heidelberg ein Büro, das von Frau Karin Dornacher geleitet wird.

## TOP 2: Bericht des Vorstands

### TOP 2.1: Tagungen

#### Zurückliegende Tagungen

- BachoT<sub>E</sub>X vom 29.4.–3.5.2009 in Bachotek in der Nähe von Brodnica.
- TUG 2009 vom 29.7.–31.7.2009 in Notre Dame, South Bend, Indiana (USA).

Es war eine relativ kleine Tagung mit ca. 50 Teilnehmern. Die Vorträge wurden live auf Video aufgenommen und sind unter der folgenden URL <http://river-valley.tv/conferences/typesetting/tug2009/> verfügbar, entweder zum Anschauen oder zum Herunterladen.

- EuroT<sub>E</sub>X 2009 vom 31.8.–4.9.2009 in den Niederlanden in der Nähe von Den Haag.

Die EuroT<sub>E</sub>X 2009 war gleichzeitig das 3. ConT<sub>E</sub>Xt Meeting und so gab es natürlich entsprechend viele Vorträge über ConT<sub>E</sub>Xt.

#### Kommende Tagungen

- GuIT Meeting, 17.10.2009 in Pisa, Italien.
- BachoT<sub>E</sub>X vom 29.4.–3.5.2010 in Bachotek in der Nähe von Brodnica.
- 4. ConT<sub>E</sub>Xt Tagung in Břejlov, ca. 40 km südöstlich von Prag.  
Der Termin steht noch nicht fest, es wird aber wohl irgendwann zwischen Ende Juli und Anfang September sein.
- TUG 2010 vom 28.6.–30.6.2010 in San Fransisco, Kalifornien (USA).  
Donald Knuth hat seine Teilnahme aus Anlass des 32-jährigen T<sub>E</sub>X Jubiläums zugesagt. Informationen dazu sind unter <http://www.tug.org/tug2010/> verfügbar.
- EuroT<sub>E</sub>X 2010 Ende August in Pisa, Italien.
- GUTenberg Tagung, November 2010 in Paris

#### DANTE e.V. Tagungen

- Frühjahrstagung in Dortmund.  
Die Tagung wird von der Firma QuinScape zusammen mit der Universität Dortmund organisiert.
- Herbsttagung 2010 voraussichtlich in Trier.
- Frühjahrstagung 2011 voraussichtlich in Bremen.

## TOP 2.2: Sonstige Veranstaltungen

- 11. Chemnitzer Linuxtage vom 14.3.–15.3.2009
- Linuxtag in Berlin vom 24.6.–27.6.2009

## TOP 2.3: Sonstiges

- T<sub>E</sub>X Live, T<sub>E</sub>X Collection  
Erscheinungsdatum der T<sub>E</sub>X Collection DVD wird voraussichtlich im November 2009 sein.
- Pressedistribution  
Die Post hat ihr Verfahren geändert. Die Sendungen müssen jetzt mit einem Barcode aus Adressangaben versehen sein, die aus der derzeitigen Mitglieder-Datenbank nicht generierbar sind. Deshalb bitte Adressänderungen unbedingt an das Büro melden.
- DSL im Büro Heidelberg  
Ein neues billigeres T-Home-Paket löst das alte DSL-Paket ab.
- Webserver  
Die DANTE e.V. Webseite hat ein neues Layout, erstellt mit dem Open Source Content Management System Apache Lenya.  
Es gibt jetzt auch ein DANTE e.V.-Wiki unter <http://projekte.dante.de/>. Auf <http://projekte.dante.de/Main/DanteRegistration> kann man sich registrieren lassen und dann aktiv an der Gestaltung des Wikis mitarbeiten.
- T<sub>E</sub>X-Stammtisch  
Die Informationen zum T<sub>E</sub>X-Stammtisch sind auf <http://projekte.dante.de/Stammtische/WebHome> zu finden. Die Kontaktpersonen der jeweiligen T<sub>E</sub>X-Stammtische können ihre Daten im Wiki selber ändern, wenn sie sich vorher registriert haben.
- T<sub>E</sub>X Projektberatung  
Martin Sievers will ein Netzwerk von T<sub>E</sub>X Ansprechpartnern aufbauen, die sich nach aussen gemeinsam präsentieren und auch Projektberatung bieten. Interessierte können sich bei ihm melden: [martin@texberatung.com](mailto:martin@texberatung.com)

Unter diesem Tagesordnungspunkt wurden keine Beschlüsse gefasst.

Klaus Höppner schließt die Versammlung um 10:50 Uhr.

Klaus Höppner  
(Versammlungsleiter)

Manfred Lotz  
(Protokollant)

# T<sub>E</sub>X-Theatertage

---

## DANTE e.V. in Esslingen

Dominik Wagenführ

### Begrüßung und Mitgliederversammlung

Nach der Registrierung begann kurz nach 9 Uhr die 41. Mitgliederversammlung bei der der Vorsitzende Klaus Höppner zuerst die Teilnehmer zur Veranstaltung begrüßte. Sehr ausführlich berichtete er danach über die verschiedenen T<sub>E</sub>X-Tagungen, die weltweit in diesem Jahr stattfanden. So erfuhr man, wie die TUG 2009 in Notre Dame, Indiana, USA, verlaufen ist und sah auch einige Bilder des schönen Campus. Auch von der EuroT<sub>E</sub>X 2009 und dem 3. ConT<sub>E</sub>Xt-Meeting, die dieses Jahr in den Niederlanden stattfanden, wurde berichtet. Neben dem Hinweis auf kommende Tagungen gab es auch einen Rückblick auf die verschiedenen deutschen Linux-Veranstaltungen wie die Chemnitzer Linuxtage und den Linuxtag in Berlin, auf denen DANTE e.V. mit Ständen und Vorträgen vertreten war.

Bei dieser Gelegenheit habe ich auf zwei L<sup>A</sup>T<sub>E</sub>X-Workshops am 17. Oktober 2009 hingewiesen, die ich im Rahmen der Ubucon an der Universität in Göttingen halten werde.

Im Anschluss wurden verschiedene Interna besprochen, vor allem aber Infos zur Zeitschrift »Die T<sub>E</sub>Xnische Komödie« wurden behandelt. Dabei gab es auch von Seiten der Teilnehmer einige Verbesserungsvorschläge und Wünsche. So würde ein Teilnehmer gerne mehr einsteigerfreundliche Artikel in der Komödie sehen. Hier trifft aber zu, was alle Magazine trifft, die keine Festangestellten haben: Es muss sich erst einmal ein Autor finden, der etwas dazu schreibt.

### Projekt: PostScript und PDF

Nach einer halbstündigen Pause gab es den ersten Vortrag. Herbert Voß, Autor vieler hilfreicher L<sup>A</sup>T<sub>E</sub>X-Lehrbücher, die auf der Tagung auch zum Kauf angeboten wurden, stellte ein Projekt vor, bei dem es darum geht, PostScript-Grafiken,

die bisher nur über den Umweg `latex`→`dvips`→`ps2pdf`→`includegraphics` in PDF-Dokumenten genutzt werden konnten, direkt mit `pdflatex` übersetzen zu können. Mittels des Perl-Programmes `pst2pdf` werden Präambel, Text und Grafik eines  $\text{\LaTeX}$ -Dokumentes aufgespalten, die Grafiken einzeln konvertiert und am Ende alles wieder zu einem Dokument zusammengefügt, aus dem man die PDF-Datei erstellen kann.

Das Programm hat dabei aber noch Einschränkungen. Zum Beispiel ignoriert es alle lokalen Änderungen für Umgebungen wie `pspicture`, die außerhalb dieser definiert werden und auch nicht global in der Präambel gegeben sind. Darüber hinaus gibt es Probleme, wenn solche Grafiken in Verbatim-Umgebungen eingebettet sind. Von einer neu definierten Umgebung auf Basis der `pspicture`-Umgebung sollte man daher wohl auch Abstand nehmen.

Ingesamt ist das aber ein spannendes Vorhaben, welches man auch in Zukunft beachten sollte, nimmt es doch einige Hürden, die einem  $\text{\LaTeX}$ -Anwender heute noch in den Weg gestellt werden, wenn man schöne Grafiken erstellen will.

## Biber - Der passende $\text{\BibTeX}$ -Nachfolger für `biblatex`

Martin Sievers stellte im folgenden Vortrag `Biber` vor, welches als Nachfolger von  $\text{\BibTeX}$  angesehen werden kann. Mittels  $\text{\BibTeX}$  und `biblatex` kann man sehr leicht Literaturdatenbanken erzeugen und schnell in einem Dokument referenzieren. Der Nachteil ist, dass  $\text{\BibTeX}$  vor allem bei der Sortierung von in Unicode kodierten Quellen (also auch deutschen Umlauten) Probleme hat und diese falsch sortiert. Daneben kann es bei großen Literaturdatenbeständen zu einem Überlauf des Speichers kommen. All dies soll mit `Biber` behoben werden.

Dazu gibt es auch die neue auf XML-basierte Sprache `Bib $\text{\LaTeX}$ XML`, mit der man die Datenbankeinträge in einer XML-Sprache leichter beschreiben kann.

## Graphviz-DOT

Nach der Mittagspause stellte Michael Niedermair das Programm `Graphviz` vor, mit dem man anhand von einfachen Regeln leicht Strukturdiagramme, Pfeildiagramme, MindMaps und Netzpläne erstellen kann. Dabei gibt man nur die einzelnen Zuordnungen der Knoten und Verbindungen an, sowie auf welche Art die Grafik erzeugt werden soll.

Aufgrund einiger Einschränkungen bei der grafischen Darstellung (keine besonderen Linien, kein Mathemodus als Beschriftung, keine Schriftarteinbindung) wurde auch der Export mittels `dot2tex` dargestellt. Über dieses kann man ein

erstelltes Diagramm in  $\text{T}_{\text{E}}\text{X}$ -Code exportieren und dann mittels des `PSTricks`-Paketes verändern.

## Dynamische Bücher

Im Anschluss folgte ein sehr interessanter Vortrag von André Wobst. Für die Erstellung von Tagungsbänden wurde ein Online-Programm erstellt, das es erlaubt, einzelne Beiträge zu selektieren und diese nach eigenen Sortiermethoden als PDF-Dokument auszugeben. Dabei werden die Texte aus einer Datenbank genommen und in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Code abgelegt, der dann mittels `HEVEA` in ein einfaches HTML-Format konvertiert oder eben per `pdflatex` übersetzt werden kann.

Eine Anforderung ist dabei, dass der `TEX2HTML`-Konverter sehr schnell arbeitet, da bei einem Tagungsband mit mehreren Tausend Beiträgen nicht erst Stunden auf das Online-Ergebnis gewartet werden kann. `HEVEA` soll dabei diese Anforderung erfüllen.

Meine persönlichen Erfahrungen mit `HEVEA` zeigen leider ein anderes Bild, was die allgemeine Anwendung angeht. Die Alternativen `tex4ht` und `tth` sind hier aber auch nicht viel besser und funktionieren oft gar nicht oder nur eingeschränkt. (Das hoffnungslos veraltete `latex2html` bleibt außen vor.)

## Komfortable Makefiles

Danach folgte ein weiterer Vortrag von Michael Niedermair, der die Verwendung von Makefiles für  $\text{T}_{\text{E}}\text{X}$ -Dokumente erklärte. Dabei wird anhand verschiedener Regeln ein  $\text{T}_{\text{E}}\text{X}$ -Dokument nur dann kompiliert, wenn sich Inhalte geändert haben. Das Ganze kann man auch so erweitern, dass beispielsweise Grafiken aus dem Programm `dia` bei Änderungen korrekt in PDF konvertiert und dann im  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Dokument eingebunden werden.

Nach der ersten Hälfte der Zeit wurde der Vortrag aber sehr technisch und durch die zahlreichen Beispiele zu ausführlich, sodass dann kaum noch etwas Neues mitgenommen werden konnte.

## Having Fun with $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ II

Nach der Kaffeepause stellte Adelheid Grob einige interessante  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Pakete vor, die die meisten Zuhörer sicherlich noch nicht kannten. Neben sehr speziellen Paketen wie `cooking`, mit dem man Rezepte setzen kann, oder `guitar`, welches für die Akkorde über Liedtexten zuständig ist, wurden auch sehr

hilfreiche Pakete wie `pseudocode` für die Darstellung von einfachen Codebeispielen und `uebungsblatt` zur Erstellung von Übungsblättern und Klausuren vorgestellt.

## Was gibt es Neues bei T<sub>E</sub>X Live?

Den Abschluss bildete Klaus Höppner, der einen Vortrag von Karl Berry zum aktuellen Stand von T<sub>E</sub>X Live 2009 zum Besten gab, welches im Herbst 2009 erscheinen soll.

Dabei beschränken sich die Änderungen gegenüber der Version 2008 auf Kleinigkeiten. So werden mittels `epstopdf` EPS-Grafiken in L<sup>A</sup>T<sub>E</sub>X-Dokumenten, die mit `pdflatex` übersetzt werden, automatisch nach PDF konvertiert.

Mit `writel8` kann man externe Befehle aus T<sub>E</sub>X heraus aufrufen, wobei aufgrund von Sicherheitsregeln aber nur bestimmte Kommandos ausgeführt werden können. (Man kann diesen Sicherheitsmechanismus aber sehr leicht aushebeln.)

Vorgestellt wurde auch der neue minimalistische T<sub>E</sub>X-Editor T<sub>E</sub>Xworks, der einen eingebauten PDF-Betrachter besitzt. Man kann mit ihm auf diese Art im PDF-Dokument eine Stelle markieren und auf den zugehörigen T<sub>E</sub>X-Quellcode im Editor springen, was sicherlich in manchen Fällen eine hilfreiche Funktion ist.

## Schlussbemerkung

Die Veranstaltung war sehr locker, alle waren freundlich und es gab viele interessante Vorträge. Am Freitagabend und am Samstagabend traf man sich in verschiedenen Restaurants, wobei man sich auch über andere Themen als L<sup>A</sup>T<sub>E</sub>X unterhalten konnte – was natürlich auf der Tagung selbst auch möglich war. Am Sonntag gab es noch eine touristische Führung durch Esslingen.

Wer an L<sup>A</sup>T<sub>E</sub>X interessiert ist und bei einer der nächsten Tagungen in der Nähe ist (die nächsten werden in Dortmund, Trier und Bremen stattfinden), sollte sich unbedingt dort anmelden. Auch wenn man nicht Mitglied von DANTE e.V. ist, kann man bei den Vorträgen zuhören und einiges an Wissen mitnehmen.

## Kieler Linux- und Open-Source-Tage am 2. und 3. Oktober 2009

Karlheinz Geyer, Adelheid Grob

### Heidi allein in Kiel

»Kiel Hauptbahnhof, Endstation. Bitte alle aussteigen!« – fast auf den Tag genau vor vier Jahren wurde ich mit den gleichen Worten in der Schleswig-Holsteinschen Landeshauptstadt begrüßt. Damals voller Aufregung – ging es doch zum ersten Mal für mich zum Vorabendtreff einer Tagung von DANTE e.V. mit lauter mir bis dato unbekanntem Gesichtern. Vier Jahre später war deutlich weniger Aufregung im Spiel, eher zum einen Kopfweg und Müdigkeit als Nachklang einer eben überstandenen heftigen Herbstgrippe und zum anderen natürlich Vorfreude. Vorfreude auf meine Freundin, die inzwischen in Kiel studiert und die ich lange nicht mehr gesehen hatte, zum anderen auf Karlheinz und den Linuxtag in Kiel. Anders als beim ersten Besuch lag meine Unterkunft diesmal auch nicht am Ende der Welt bzw. am anderen Ende Kiels, sondern am schönen Hindenburgufer direkt an der Kieler Förde zwischen Nobelvillen rund um den Landtag.

Nachdem ich am Donnerstag abend eine Stadtführung sowie ein sehr leckeres Essen bekam, war ich am ersten Tag der Kieler Linux- und Open-Source-Tage auch bestens gerüstet für die Aufgabe, DANTE e.V. diesmal nicht selber näher kennen zu lernen, sondern anderen zu erklären, was unser Verein denn so tut und was  $\text{\LaTeX}$  denn überhaupt ist.

### Zu zweit auf den Linuxtagen

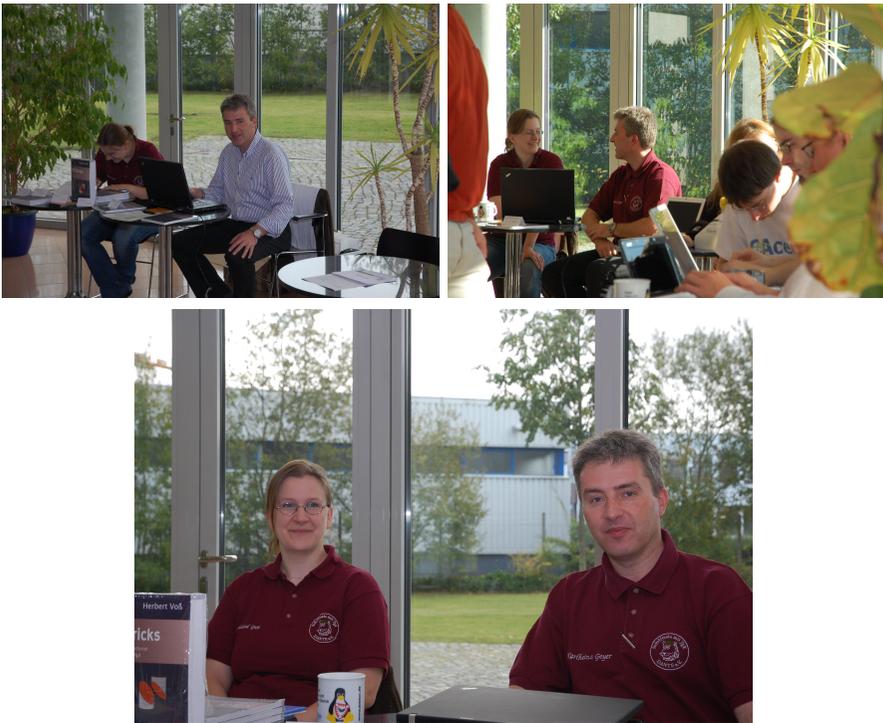
Erwartungsvoll kamen Karlheinz (frisch eingeflogen) und Heidi am Freitag morgen dann im beherbergenden Institut (KITZ) an und waren sehr erstaunt ob der Größe des Linuxtages - insgesamt waren ein paar mehr als zehn Aussteller da und unser »Stand« bestand aus einem Tisch sowie zwei Stühlen. Dafür gab es Strom, sofort einen netten Ansprechpartner und – ganz wichtig – eine Kaffeeplatte! Kurzerhand wurde noch ein zweiter Tisch und Strom organisiert und los konnte es gehen . . .

Trotz oder gerade wegen der geringen Größe der Veranstaltung, war diese dennoch sehr erfolgreich: Zum einen trafen am Stand ein paar Teilnehmer der Tagung vor vier Jahren ein, zum anderen kamen etliche Fragen von Anwendern, denen geholfen werden konnte. »Neulingen« konnte erklärt werden, was man

mit L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt oder ähnlichem alles machen kann. Es gelang außerdem, Sören – ein Teilnehmer der 2005er Tagung – und noch ein paar andere, von der Idee zu begeistern, den Kieler Stammtisch wiederzubeleben. Schon auf den Linuxtagen fanden sich hierfür vier bis fünf Begeisterte, die einen Termin für ein erstes Treffen koordinieren wollten. Von den von Frau Dornacher nach Kiel geschickten Bücher konnten bis auf zwei alle verkauft werden, auch etliche DVDs der T<sub>E</sub>XLive fanden Abnehmer und ein Schnuppermitglied konnte geworben werden. Als wesentlich lässt sich festhalten, dass unser Stand meistens von einem oder mehreren Interessierten umringt war. Fazit: Eine Teilnahme im nächsten Jahr ist wieder zu empfehlen – dann vielleicht unterstützt oder ausgerichtet von Mitglieder des Kieler Stammtisches!

### Abschliessende Bemerkung von Heidi

Für die, die es immer interessiert: Ich habe aufgrund des miesen Wetters leider nur zwei »Geocaches« finden können – nachdem ich am Samstagfrüh total durchnässt war, musste ich leider aufgeben. . .



# Bretter, die die Welt bedeuten

---

## Aufgabenblätter, Tests, Klausuren, . . .

Jürgen Gilg

Das `eqexam`-Paket bietet die Möglichkeit, Aufgabenblätter, Tests und Klausuren einfach zu setzen und die jeweils zugehörigen Lösungen mit in die jeweiligen Aufgabenstellungen zu integrieren. Es werden die beiden einfachen Umgebungen `\begin{problem} . . . \end{problem}` und deren Sternversion `\begin{problem*} . . . \end{problem*}` vorgestellt, um daran zu zeigen, wie man mit den zugehörigen Lösungen der Aufgaben einfach jonglieren kann. Des Weiteren wird auch die Bewertung der Aufgaben mit Punkten und die einfache Handhabung von Querverweisen zu den erstellten Aufgaben oder Teilaufgaben diskutiert.

## Bevor man lostippt . . .

### Installation des Pakets

Erstellen Sie einen Ordner im Suchpfad Ihres  $\text{T}_{\text{E}}\text{X}$ -Systems, indem Sie `eqexam.zip` dorthin kopieren und entpacken. Öffnen Sie anschließend `eqexam.ins` in Ihrem Editor und führen Sie einen  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Lauf aus, um die Dateien `eqexam.sty` und `eqalone.def` zu generieren.

`eqexam` ist ein Teil des *AcroT<sub>E</sub>X eDucation Bundle*, kann aber auch alleine genutzt werden (<http://mirror.ctan.org/macros/latex/contrib/eqexam/>). Die Datei `eqalone.def` enthält die dafür notwendigen Definitionen und Befehle. Um den vollen Umfang des Pakets zu nutzen wird empfohlen, die zwei Pakete `web` und `exerquiz` vom *AcroT<sub>E</sub>X eDucation Bundle* ebenfalls zu laden.

### Notwendige Pakete und unterstützte Treiber

`eqexam` lädt folgende Pakete, die möglicherweise nicht standardmäßig in Ihrem  $\text{T}_{\text{E}}\text{X}$ -System installiert sind:

- `calc`, um die Positionen der Punktzahlen der Aufgaben am Rand des Dokuments zu berechnen,
- `pifont`, für Symbole, die bei Nutzung der `proofread`-Option in den Auswahlkästchen bei multiple-choice-Fragen stehen,
- `comment`, für verschiedene Versionen eines Aufgabenblatts,
- `multicol`, für mehrspaltiges Setzen der Aufgaben oder Teilaufgaben,
- `verbatim`, zum Schreiben der Hilfsdateien auf die Festplatte,
- `amstext`, `amssymb`, für den Formelsatz,
- `xcolor`, für das Farbmanagement bei Online-Aufgaben.

Die vom Paket unterstützten Treiber sind `dvips`, `dvipson`, `pdftex` und `dvipdfm`.

### Die wichtigsten Hilfsdateien

Im ersten  $\text{\LaTeX}$ -Lauf werden folgende Hilfsdateien erstellt:

- `\jobname.sol`: Diese beinhaltet sämtliche Lösungen der Aufgaben und Teilaufgaben.
- `\jobname.aux`: Das ist die standardmäßige, von  $\text{\LaTeX}$  erzeugte Hilfsdatei.
- `comment.cut`: Dies ist eine spezielle Hilfsdatei, die notwendig ist, wenn verschiedene Versionen eines Aufgabenblatts erstellt werden sollen.

Die Handhabung der zugehörigen Lösungen wird mittels der Hilfsdatei `\jobname.sol` erledigt. Im ersten  $\text{\LaTeX}$ -Lauf wird diese erstellt und im zweiten Lauf an unterschiedlichen Stellen im Dokument wieder eingebunden.

## Zwei einfache Umgebungen

### Die `problem`-Umgebung

Diese Umgebung eignet sich für einfache Aufgaben ohne Teilaufgaben. Die Struktur ist:

```

1 \begin{problem}
2 Hier kommt der Text der Aufgabe.
3 \begin{solution}
4 Die Lösung zur Aufgabe
5 \end{solution}
6 \end{problem}

```

Diese Umgebung hat zwei optionale Argumente. Das erste ist:

- `<num>`: Eine Zahl, beispielsweise `\begin{problem}[5]`; diese Zahl 5 gibt die gewünschte Punktzahl der Aufgabe an.
- `<empty>`: Wenn man auf Punkte verzichten möchte, lässt man das erste optionale Argument leer `\begin{problem}[]` oder weg `\begin{problem}`.

Das zweite optionale Argument ist im Rahmen dieser kleinen Einführung ohne Belang und soll daher hier nicht diskutiert werden. In jede `problem`-Umgebung ist direkt die `solution`-Umgebung integriert, in die man die Lösung der Aufgabe eingibt. Man sieht nun sehr leicht die Übersichtlichkeit der Aufgabe, die direkt von ihrer zugehörigen Lösung gefolgt wird.

### Die `problem*`-Umgebung

Die `problem*`-Umgebung ist für eine Aufgabe mit mehreren Teilaufgaben (a), (b), (c), ... gemacht. Die Teilaufgaben werden in eine `parts`-Umgebung gesetzt und mit `\item` aufgelistet. Die Struktur ist:

```

1 \begin{problem*}
2 Hier kommt allgemeiner Text.
3 %% Start der Teilaufgaben
4 \begin{parts}
5 \item Die erste Teilaufgabe
6 \begin{solution}
7 Die Lösung zur ersten Teilaufgabe
8 \end{solution}
9 \item Die zweite Teilaufgabe
10 \begin{solution}
11 Die Lösung zur zweiten Teilaufgabe
12 \end{solution}
13 [ ... ]
14 \end{parts}
15 \end{problem*}

```

Man sieht, dass auch hier die Lösungen zu den Teilaufgaben direkt den jeweiligen Teilaufgaben folgen. Die `problem*`-Umgebung hat zwei optionale Argumente. Das erste optionale Argument ist eines der folgenden vier Parameter:

- `<num>`: Ist das erste Argument der Umgebung `problem*` eine Zahl, beispielsweise `\begin{problem*}[5]`, so gibt diese die Gesamtpunktzahl aller Teilaufgaben an. Hier wird also keine Gewichtung der Teilaufgaben vorgenommen.
- `<num>ea`: Ist das erste Argument eine Zahl direkt gefolgt von »ea« (eine Abkürzung für *each*), z. B. `\begin{problem*}[3ea]`, so wird jede Teilaufgabe mit jeweils 3 Punkten gleich gewichtet.
- `\auto`: Ist das erste Argument `[\auto]`, dann wird die Gesamtpunktzahl der Aufgabe durch die Summe der Punkte der einzelnen Teilaufgaben errechnet. Die Einzelpunktzahlen der Teilaufgaben werden wie folgt angegeben: Eine mit `\item\PTs{3}` eingeleitete Teilaufgabe trägt 3 Punkte, eine Teilaufgabe mit `\item\PTs{5}` trägt 5 Punkte zur Gesamtpunktzahl der Aufgabe bei. Ein einfaches Beispiel:

```

1 \begin{problem*}[\auto]
2 Hier ein wenig Text.
3 \begin{parts}
4 \item\PTS{3} Die 3-Punkte-Teilaufgabe
5 \begin{solution}
6 Die Lösung der ersten Teilaufgabe.
7 \end{solution}
8 \item\PTS{4} Die 5-Punkte-Teilaufgabe
9 \begin{solution}
10 Die Lösung der zweiten Teilaufgabe.
11 \end{solution}
12 \end{parts}
13 \end{problem*}

```

Dies liefert also eine Aufgabe mit zwei integrierten Teilaufgaben und der Gesamtpunktzahl  $3 + 5 = 8$ .

- `<empty>`: Wenn man auf Punkte verzichten möchte, lässt man das erste optionale Argument leer: `\begin{problem*}[]`. Damit ist auch das noch zu diskutierende zweite optionale Argument überflüssig und man kann vereinfachend beide optionale Argumente weglassen, also `\begin{problem*}` schreiben.

Das zweite optionale Argument `[\Do<num>]` wird nur in Kombination mit gleichgewichteten Teilaufgaben verwendet, also beispielsweise mit

`\begin{problem*}[5ea][\Do3]`.

Dies ermöglicht sogenannte Auswahlfragen. Wenn man z. B. vier Teilaufgaben hat – alle mit 5 Punkten gleichgewichtet (`[5ea]`) – und man zusätzlich noch `[\Do3]` als zweites optionales Argument anhängt, dann hat die Aufgabe eine Gesamtpunktzahl von  $3 \cdot 5 = 15$  Punkten und der Schüler soll nach seiner eigenen Auswahl drei der vier gestellten Teilaufgaben behandeln.

## Möglicher Aufbau eines einfachen Aufgabenblattes

In diesem Abschnitt wird gezeigt, wie man diese einzelnen Aufgabentypen zu einem Aufgabenblatt zusammenstellt.

```

1 \documentclass{article}
2
3 \usepackage[myconfigi,pointsonleft,forpaper]{eqexam}
4 \subject[DANTE]{DANTE} %% Für die Kopfzeile
5 \title[T1]{Test 1} \author{Jürgen Gilg} \date{WS, \the\year}
6
7 %% Wichtig für die _solutiononly_ Option
8 %\encloseProblemsWith{unique_name}
9 %% Individuelle Umbenennungen. Kommen aus der myconfigi.cfg
10 % \examNameLabel{Name;}

```

```

11 % \ptsLabel{Pkt}
12 % \eachLabel{jew.}
13 % \pointsLabel{Punkte}
14
15 \begin{document}
16 \maketitle
17 \begin{exam}{myProblems}
18 %% Wichtig für die _solutiononly_ Option
19 %\begin{unique_name}
20
21 \begin{instructions}[Anleitung]
22 Dies ist eine Anleitung zu den zu bearbeitenden Aufgaben.
23 \end{instructions}
24 \begin{problem}[5] %% einfache Frage
25 Dies ist eine einfache Frage (ohne Teilfragen).
26 \begin{solution}
27 Dies ist die zugehörige Lösung.
28 \end{solution}
29 \end{problem}
30 \begin{problem*}[3ea]%% Aufgabe mit drei Teilfragen (mit der *-Version ←
    aufgerufen)
31 Dies ist eine Aufgabe mit folgenden Teilfragen:
32 \begin{parts}%% Hier starten die Teilfragen (mit \item gelistet)
33 \item Dies ist die erste Teilfrage.
34 \begin{solution}
35 Dies ist die Lösung zur ersten Teilfrage.
36 \end{solution}
37 \item Dies ist die zweite Teilfrage.
38 \begin{solution}
39 Dies ist die Lösung zur zweiten Teilfrage.
40 \end{solution}
41 \item Dies ist die dritte Teilfrage.
42 \begin{solution}
43 Dies ist die Lösung zur dritten Teilfrage.
44 \end{solution}
45 \end{parts}
46 \end{problem*}%% Hier enden die Teilfragen
47
48 %\end{unique_name}%% Wichtig für die _solutiononly_ Option
49 %\includeexersolutions
50 \end{exam}
51
52 \end{document}

```

So hat man nun auf übersichtliche Weise ein Aufgabenblatt mit Fragen und zugehörigen Lösungen (im Quellcode direkt untereinander). Das erzeugte Ergebnis ist:

DANTE WS, 2009	Test 1	Name: _____ Jürgen Gilg
<b>Anleitung</b> (14 Punkte) Dies ist eine Anleitung zu den zu bearbeitenden Aufgaben.		
(5 <sup>Pkt</sup> )	1. Dies ist eine einfache Frage (ohne Teilfragen).	
(3 <sup>Pkt</sup> (3 <sup>sew.</sup> ))	2. Dies ist eine Aufgabe mit folgenden Teilfragen:	
	(a) Dies ist die erste Teilfrage.	
	(b) Dies ist die zweite Teilfrage.	
	(c) Dies ist die dritte Teilfrage.	

Abbildung 1: Aufgabenblatt

	Seite 2 von 2
<b>Lösungen zu DANTE</b>	
1. Dies ist die zugehörige Lösung.	<input type="checkbox"/>
2. (a) Dies ist die Lösung zur ersten Teilfrage.	<input type="checkbox"/>
2. (b) Dies ist die Lösung zur zweiten Teilfrage.	<input type="checkbox"/>
2. (c) Dies ist die Lösung zur dritten Teilfrage.	<input type="checkbox"/>

Abbildung 2: Lösungen zu den Aufgaben

*Hinweis:* Das Kopfzeilenlayout wurde aus der im Paket beiliegenden Konfigurationsdatei `eqexam.cfg` kopiert, leicht modifiziert und in einer eigenen Konfigurationsdatei `myconfig1.cfg` gespeichert. Diese wird auf die eigenen Bedürfnisse angepasst und mit der Einstellung `\usepackage[myconfig1]{eqexam}` geladen (dies ist für den Namen `myconfig1.cfg` im Paket schon vorbereitet). Natürlich kann man auch eine `myname.cfg` erstellen und diese mit `\input myname.cfg` in der Präambel laden.

## Querverweise

Man kann auch auf die Aufgaben/Teilaufgaben verweisen, wie man das mit  $\LaTeX$  gewohnt ist. Dies illustriert folgendes kleines Beispiel:

```

1 \begin{problem}[5]\label{probl}
2 Dies ist eine einfache Frage (ohne Teilfragen).
3 \begin{solution}
4 Dies ist die zugehörige Lösung.
5 \end{solution}
6 \end{problem}
7 \begin{problem*}[3ea]
```

```

8 Dies ist eine Aufgabe mit folgenden Teilfragen:
9 \begin{parts}
10 \item Dies ist die erste Teilfrage.
11 \begin{solution}
12 Dies ist die Lösung zur ersten Teilfrage.
13 \end{solution}
14 \item\label{teilprob2} Dies ist die zweite Teilfrage.
15 \begin{solution}
16 Dies ist die Lösung zur zweiten Teilfrage.
17 \end{solution}
18 \end{parts}
19 \end{problem*}
20
21 Hier noch ein Beispiel des einfachen Referenzierens.
22
23 Bitte bearbeiten Sie Aufgabe~\ref{probl} und Teilaufgabe~\ref{teilprob2}.

```

DANTE WS, 2009	Test 1	Name: _____ Jürgen Gilg
<b>Anleitung</b> (14 Punkte) Dies ist eine Anleitung zu den zu bearbeitenden Aufgaben.		
(5 <sup>Pkt</sup> )	1. Dies ist eine einfache Frage (ohne Teilfragen).	
(9 <sup>Pkt</sup> <sub>3ew.</sub> )	2. Dies ist eine Aufgabe mit folgenden Teilfragen:	
	(a) Dies ist die erste Teilfrage.	
	(b) Dies ist die zweite Teilfrage.	
	(c) Dies ist die dritte Teilfrage.	
Hier noch ein Beispiel des einfachen Referenzierens. Bitte bearbeiten Sie Aufgabe 1 und Teilaufgabe 2(b).		

Abbildung 3: Querverweise zu den Aufgaben

## Jonglieren mit den Lösungen der Aufgaben/Teilaufgaben

Nun kommen wir zu einer großen Stärke des `exexam`-Pakets – die optionalen Argumente des Pakets selbst. Bezug genommen wird auf das vorher vorgestellte Beispiel eines Aufgabenblatts. Ich möchte hier vier optionale Argumente vorstellen, die ein und dasselbe Aufgabenblatt unterschiedlich ausgeben lassen können.

- `<empty>`: Ohne optionales Argument des Pakets werden nun alle Fragen samt Teilfragen untereinander nummeriert angeordnet, und die zugehörigen Lösungen werden nach allen Fragen der Reihenfolge nach ebenfalls untereinander nummeriert angeordnet – begonnen auf einer neuen Seite.

- `nosolutions`: Bei Wahl des optionalen Arguments `nosolutions`, also `\usepackage[nosolutions]{eqexam}`, werden keine Lösungen ausgegeben.
- `solutiononly`: Bei Wahl des optionalen Arguments `solutiononly`, also `\usepackage[solutiononly]{eqexam}`, werden keine Aufgaben (und Teilaufgaben), sondern nur die Lösungen gezeigt. Wichtig zu beachten bei dieser Option: `\encloseProblemsWith{unique_name}` (Zeile 9), `\begin{unique_name}` (Zeile 22), `\end{unique_name}` (Zeile 58) und `\includeexersolutions` (Zeile 59) müssen auskommentiert werden und vorab mit der `nosolutions` Option übersetzt werden, um die zugehörige Datei `\jobname.sol` zu erzeugen. Erst danach ist die `solutiononly` Option zu wählen.
- `answerkey`: Bei Wahl dieses optionalen Arguments beim Laden des Pakets, also `\usepackage[answerkey]{eqexam}`, werden die Lösungen direkt unter den Fragen bzw. Teilfragen angeordnet. Dies ist sehr gut zum Korrekturlesen der Aufgaben mit den zugehörigen Lösungen geeignet.

*Hinweis:* Es wird das `calc`-Paket genutzt und dies erfordert einen dreimaligen  $\LaTeX$ -Lauf, damit die Positionen für die Punktzahlen korrekt übergeben und alle Daten aus den Hilfsdateien sauber erstellt werden. Die Summe der Punktzahlen wird mit  $\TeX$ -Routinen erzeugt.

Ein nützlicher Trick ist, sämtliche gewünschten Optionen untereinander mit `»%«` in die Präambel zu kommentieren und dann die jeweils gewollte Option zu aktivieren.

```

1 \usepackage[%
2   %nosolutions%
3   %solutiononly%
4   %answerkey%
5   ]{eqexam}

```

Für die Handhabung der Punktzahlen (mit oder ohne Gesamtsumme aller Aufgaben, mit seitenweiser Teilsumme usw.) gibt es weitere optionale Argumente, die selbsterklärend sind: `pointsonleft`, `pointsonright`, `pointsonboth`, `nopoints`, `totalsonleft`, `totalsonright`, `nototals`, `noparttotals`, `nosummarytotals`.

## Was nicht vorgestellt wurde ...

Das `eqexam`-Paket von D. P. Story ist sehr viel umfangreicher: Multiple-choice-Aufgabentypen, Mehrfachauswahl bei multiple-choice-Aufgabentypen, fill-in-the-blank-Aufgabentypen, true/false-Aufgabentypen, individuelle Titelseiten- und Kopfzeilengestaltung, Erzeugen eines Aufgabenblattes für Papier (auch für mehrere Versionen des Aufgabenblatts – Gruppe A oder Gruppe B oder

Gruppe  $N \leq 26$ ) oder Onlinebearbeitung (aus der Papierversion wird eine interaktive Onlineversion durch Änderung der Optionen) und sehr vieles mehr. Ein Blick in die in englischer Sprache geschriebene und ausführliche Dokumentation lohnt sich für jeden, der sich mit dem Setzen von Aufgaben für Hausaufgabenblätter, Tests oder Klausuren samt den zugehörigen Lösungen beschäftigt. Eine aktuelle Version des Pakets wird Ende 2009 auf CTAN verfügbar sein. Die jeweils aktuelle Version finden Sie unter: <http://www.math.uakron.edu/~dpstory/eqexam.html>

Ein Musterbeispiel, das auch die erweiterten Möglichkeiten des Pakets aufzeigt, ist im Paket enthalten und findet sich ebenfalls auf der oben genannten Website. Die optionalen Pakete `web` und `exerquiz` finden Sie unter: <http://www.math.uakron.edu/~dpstory/webeq.html>

*Hinweis:* Sollte jemand spezielles Interesse am Erstellen von interaktiven Onlinetests haben, dann lohnt sich in jedem Fall ein Blick in die Dokumentation des *AcroT<sub>E</sub>X eEducation Bundle*. Dies unterstützt dieselben Treiber wie `eqexam`, jedoch wird der Adobe Reader als pdf-Viewer benötigt, um das im Paket (und damit auch im pdf) implementierte JavaScript auszuführen. Dies hat den großen Vorteil, dass die erstellten pdf-Dokumente auch Offline interaktiv sind. Die Aufgabeneingabe im Quellcode ist den in `eqexam` vorgestellten Umgebungen strukturell gleich: [http://www.math.uakron.edu/~dpstory/acrotex/aeb\\_man.pdf](http://www.math.uakron.edu/~dpstory/acrotex/aeb_man.pdf).

# OpenType Engineering in T<sub>E</sub>X

## Summary Report for the Oriental T<sub>E</sub>X Project

Idris Samawi Hamid

### Background: The Oriental T<sub>E</sub>X Project

The Oriental T<sub>E</sub>X project is an ambitious attempt to bring high quality Arabic-script typesetting to academics and, beyond, to the world at large. Recent decades have witnessed the degradation of typographical and typesetting quality of Arabic-script texts, in particular scholarly books. For example, complete sets of vowel markings and diacritics, crucial for understanding the meaning of a given text, are rarely available or readily accessible. This and other features so essential to a *culturally authentic* representation of Arabic script are woefully lacking in even the most expensive word processors and layout programs. Compared to the corresponding situation in Latin typography Arabic script resides in a veritable dark age.

For the current round of funding through DANTE e.V., we are focusing on a problem that is by no means limited to the Arabic script. It turns out that a subset of the problems Arabic-script typography can be solved in a manner that will be of equal benefit to Latin-script typography or, indeed, any other script. This is the *implementation of a flexible OpenType layout engine within T<sub>E</sub>X itself*.

### The Importance of OpenType layout for T<sub>E</sub>X

When T<sub>E</sub>X was initially developed by Donald Knuth, it became *the* standard of digital typography for a long time. As a general layout engine, T<sub>E</sub>X and its offspring such as pdfT<sub>E</sub>X—and now luaT<sub>E</sub>X—arguably remain unsurpassed in many ways. With respect to microtypography, structured typography, and automated typography, T<sub>E</sub>X still leads the way.

However, in the area of font technology, T<sub>E</sub>X has fallen behind. Two developments in particular have placed T<sub>E</sub>X in the position of having to play catchup:

- The Unicode standard for character encoding;

- The OpenType standard for the representation of both actual character shapes, on the one hand; and the layout rules to be followed by the character shapes. This representation is contained in the *OpenType font*.

Although support for the Unicode text encoding and for accessing glyphs in OpenType fonts has already existed for some time, it has mostly been rudimentary and limited by restrictions inherent in the main T<sub>E</sub>X engines of the time such as pdfT<sub>E</sub>X and Omega/Aleph. And layout implementation for advanced typography tables in OpenType fonts was all but nonexistent. X<sub>Y</sub>T<sub>E</sub>X offers access to operating system libraries, but this leaves the user and/or T<sub>E</sub>X developer only a hardwired solution that is inconsistent across operating systems and often incomplete in other ways. With a T<sub>E</sub>X-based implementation using a lua-extension layer, we have quick and easy access and control over OpenType layout behavior.

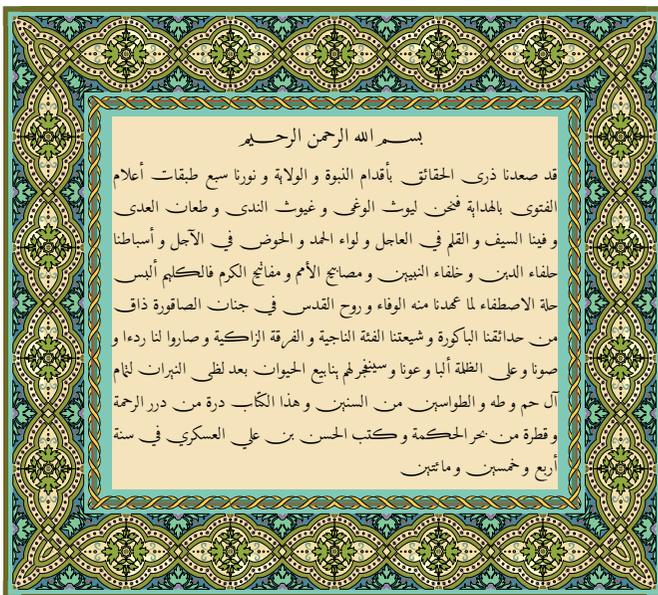


Figure 1: Opentype Control Mechanism: One example with Wide Kaaf Nuun Siin Ssaad Baa Faa Qaaf.

Early on in the Oriental T<sub>E</sub>X project, the expressive power of the Advanced Typographic or OpenType Layout tables convinced Hans Hagen, Taco Hoekwater, and myself to make OpenType the foundation for our planned implemen-

tation of culturally authentic Arabic script. Because the Arabic script is so sophisticated, one needs to use virtually every type of what is called a *lookup* offered by the OpenType specification. Thus, the development of our advanced Arabic font—codenamed “Husayni”—provides a fortuitous extreme torture test for the implementation of an OpenType layout engine for general use by users of T<sub>E</sub>X.

## The Challenges of OpenType Layout Implementation

Implementing an OpenType layout engine for T<sub>E</sub>X involves implementing the OpenType Layout Specification. See <http://www.microsoft.com/typography/otspec/>.

The OpenType Specification was developed by Microsoft and Adobe, and is still in development; the current version is 1.6. Unfortunately, the specification is quite vague in places, and neither Adobe, Microsoft, or anyone else supports all aspects of it consistently or completely. This creates lots of difficulties, and Hans Hagen and I have spent seemingly countless hours trying to sort through the vagueness. This is especially true of the specification for Global POSitioning layout (GPOS), although the Global SUBstitution (GSUB) layout-table description is not devoid of vagueness as well.

For our implementation of OpenType layout tables, we need a solid, reliable tool for adding the GSUB and GPOS tables to the font. I experimented at first with FontForge and the Adobe Feature File specification: [http://www.adobe.com/devnet/opentype/afdko/topic\\_feature\\_file\\_syntax.html](http://www.adobe.com/devnet/opentype/afdko/topic_feature_file_syntax.html).

Unfortunately, parts of the feature-file specification were unimplemented, although that has been partially rectified at the time of writing this report. But at the time of initial development, these constituted huge lacunae, and FontForge had difficulties handling feature files. Also, for anything but the simplest *lookups*, the feature file syntax becomes quite unwieldy. In particular, the syntax for specifying contextual substitutions or positionings is very opaque.

FontForge implements most of the spec, perhaps enough for everything we will need. But the preview tools are quite weak. More importantly, the GUI is far too unstable to be relied upon for hours and hours of constant use. George Williams the developer could patch things, but adding the FontForge debug-compilation cycle would add way too much overhead in terms of resources and time.

Another complication is something we have alluded to earlier: the incomplete and inconsistent implementation of OpenType layout by various applications.

As a benchmark for Hans Hagen's work on the layout engine, I experimented with applications that used Pango, Qt, ICU (International Components for Unicode, used by X<sub>Y</sub>T<sub>E</sub>X), and Uniscribe (MS Windows, used by Windows Notepad). Every one of these had problems, but Uniscribe turned out to be the most consistent and least buggy. On the other hand, Uniscribe adds a lot of script information over and above that which is contained in the OpenType specification. That is necessary for many purposes, but the MS interpretations of script needs are not canonical. More importantly, it is not always immediately obvious when, if our font fails, it is due to a bug in the font, a bug in Uniscribe, or an interpretation of Uniscribe that cancels out the feature in the font.

In addition, OpenType layout engines by especially Adobe and Microsoft implement layout with respect to, not *lookups*, but *features*. A *feature* is a tag that has a canonical general description of the typographic feature to be expected by the user assuming that the feature is turned on. The *lookup* is more low-level. A given lookup is basically a typographical function that provides a transformation unique to that lookup. So even if Adobe or Microsoft support a given class of lookups, if the lookups are collected into a feature with a name not supported by those companies, that feature won't work. So in working with, e. g., Arabic, one is restricted to a certain set of official feature tags. It is surprising that so many applications inconsistently support feature tags, since most of them represent a generic set of GSUB or GPOS operations that require virtually nothing extra by way of application support.

On the other hand, after extensive experimentation it turns out that, as of this writing, no layout engine implements the OpenType specification as consistently and as completely as Microsoft's OpenType Layout Services (OTLS), used by Uniscribe. It is very important to distinguish OTLS from Uniscribe. OTLS is the purely OpenType layout engine, to which Uniscribe adds extra script interpretation. Although OTLS is not generally available outside of Uniscribe, it is a component in the freely available VOLT (Visual OpenType Layout Tool) Microsoft's OpenType-layout development tool.

Although I would have preferred to use free tools, VOLT provides the Oriental project a relatively solid, reliable, and efficient development tool with support for both left-to-right and right-to-left scripts. The GUI generally does not crash, and the Proofing Tool uses the OTLS. Although FontForge allows one to edit in terms of the actual structure of the OpenType tables themselves, and is thus pedagogically more instructive, VOLT hides a lot of the complexity and is thus a much more efficient development environment.

VOLT is not perfect. After defining about 190 GSUB lookups, the Proofing Tool begins ignoring some later lookups. In addition, VOLT's compilation of GPOS tables starts to fail if too many anchors are defined, although I have found ways to export the anchor tables to a text file, reorganize the anchors, then get the font to compile. As simple as it may sound, this has been no easy task.

So, even before we get to T<sub>E</sub>X, OpenType-table development is a cutting edge task. The specification is often vague, and utilities are not 100% reliable.

## The ConT<sub>E</sub>Xt MkIV Implementation of an OpenType Layout Engine

In our workflow, Hans Hagen and I now use VOLT as the benchmark for the implementation of OpenType layout in T<sub>E</sub>X. This makes sense for a number of reasons, most importantly of which is the fact that more professional OpenType fonts for multilingual scripts are developed in VOLT by far than by any other tool. This includes multilingual OpenType fonts most likely to be employed by T<sub>E</sub>X users. Even Adobe professionals use VOLT, despite the presence of their own feature-file based AFDKO (Adobe Font Development Kit for OpenType).<sup>1</sup>

The first layer to the implementation of OpenType-layout tables is the use of luaT<sub>E</sub>X, whose initial iteration was funded by the original grant of the Oriental T<sub>E</sub>X project. This provides the “hooks” for an extension layer that turns luaT<sub>E</sub>X into an OpenType-layout engine. Thus, OpenType-layout is almost entirely written in Lua. The code is part of the ConT<sub>E</sub>Xt MkIV framework.

Hans Hagen and I operate iteratively: I extensively test the system by adding more and more lookups to our ever more sophisticated Arabic-script font, code-named Husayni. Then I torture the framework: each time we find a bug I implement the lookup in OTLS and visually compare it with the results of MkIV. Sometimes the bug is in MkIV, sometimes in the Husayni font, and sometimes in FontForge code used by luaT<sub>E</sub>X to open and read OpenType fonts. In addition, Hans helps with scripts to, e. g., help manipulate the text version of a VOLT file, which is then fed back into VOLT.

Often, Hans and I spend hours interpreting the VOLT results. On very rare occasions, we reject the VOLT interpretation of the spec. The one case this happened was a rather esoteric case where it made much more sense to optimize the font than add an apparently illogical hack to the implementation of successive Pair Positioning lookup types.

---

<sup>1</sup>See, e. g., [http://blogs.adobe.com/typblography/2007/06/adobe\\_arabic\\_sa.html](http://blogs.adobe.com/typblography/2007/06/adobe_arabic_sa.html)

More recently—early October this year—we found a case where FontForge was confused about how to apply the so-called “Ignore Combining Marks” flag to certain lookups. In VOLT on the contrary the flag was applied by default.<sup>2</sup>

In this case Hans and I had to bring Taco Hoekwater into the discussion and patch luaT<sub>E</sub>X to get the correct interpretation of the tables.

These kinds of patches are now rather rare. At the beginning of the development of the MkIV OpenType layout engine we had to debug luaT<sub>E</sub>X proper much more often. Now that tends to be less common, as in the case mentioned above.

A notoriously difficult case involves the combination of right-to-left and left-to-right flags in pair-positioning lookups on the one hand, with automatic unicode character mirroring on the other, in bidirectional contexts. The coordinate system for single- and pair-adjustment lookups is mirrored, relative to the current bidi flag (RTL or LTR). For characters like Arabic-script numerals, which are only “weakly” bidi, and which are generally written left-to-right, the question of whether it is more appropriate to do pair adjustments with RTL or LTR is still unsettled. VOLT can do both, but Uniscribe—which implements character mirroring—will not properly recognize the RTL solution (used by Husayni), while MkIV—with character mirroring on—will recognize the RTL solution but still has a few issues with the LTR solution (used by Microsoft’s arabtype.ttf). See `test-037a.pdf`

Hans Hagen has implemented an extensive tracking system so that we can literally “see” what happens at each stage of the lookup process. I have included a few samples of this in the appendix to this report.

In addition to Arabic-script and latin-script testing, we are also developing a test suite of geometric figures for GPOS especially. This allows one to look at the problem lookups in a more script-independent manner, so as to better focus on the real issues.

Hartmut Henkel also plays an important roles with respect to luaT<sub>E</sub>X. In particular Hartmut is continually improving bidirectional support within the luaT<sub>E</sub>X engine.

---

<sup>2</sup>See <http://fontforge.sourceforge.net/pfaeditchangelog.html>

FontForge always used to set the "ignore-combining-marks" flag bit for all ligatures. But yudit (and probably gedit) will not process the substitution if that bit is set. On the other hand I believe some arabic ligatures may depend on it (at least arabtype-volt.ttf uses it). So... now we only set this bit for arabic ligatures which do not contain combining marks. Dunno if that is right either.

## Current Support

At present ConT<sub>E</sub>Xt provides basic support for virtually all lookups given in the OpenType specification. These include

- **Glyph substitution (GSUB) rules**
  - ▷ GSUB LookupType 1—Single substitution
  - ▷ GSUB LookupType 2—Multiple substitution
  - ▷ GSUB LookupType 3—Alternate substitution
  - ▷ GSUB LookupType 4—Ligature substitution
  - ▷ GSUB LookupType 5—Contextual substitution
  - ▷ GSUB LookupType 6—Chaining contextual substitution
  - ▷ GSUB LookupType 7—Extension substitution
  - ▷ GSUB LookupType 8—Reverse Chaining Single Substitution
- **Glyph positioning (GPOS) rules**
  - ▷ GPOS LookupType 1—Single adjustment positioning
  - ▷ GPOS LookupType 2—Pair adjustment positioning
  - ▷ GPOS LookupType 3—Cursive attachment positioning
  - ▷ GPOS LookupType 4—Mark-to-Base attachment positioning
  - ▷ GPOS LookupType 5—Mark-to-Ligature attachment positioning
  - ▷ GPOS LookupType 6—Mark-to-Mark attachment positioning
  - ▷ GPOS LookupType 7—Contextual positioning
  - ▷ GPOS LookupType 8—Chaining contextual positioning
  - ▷ GPOS LookupType 9—Extension positioning

Among the above, ConT<sub>E</sub>Xt MkIV is apparently the first OpenType layout application that supports [GSUB LookupType 3] or alternate substitution. See for example, [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrssi&id=arabicfonts](http://scripts.sil.org/cms/scripts/page.php?site_id=nrssi&id=arabicfonts).

Additionally, Scheherazade includes two simplified alternates for U+06DD ARABIC END OF AYAH under the Stylistic Alternates (`salt`) feature, but at this time we know of no OpenType-based applications that can access these.

Our current system processes the alternate-substitution lookups represented by Scheherazade’s `salt` feature quite handily. This illustrates the point that MkIV has already moved ahead of the game in some respects.

## A Real-life Application: Chapter 1 of the Quran

I am happy to announce that, as of now, the Oriental T<sub>E</sub>X project, with the help of DANTE e.V. and ConT<sub>E</sub>Xt MkIV, is now capable of typesetting Quran with at least an order of magnitude greater flexibility and beauty than just

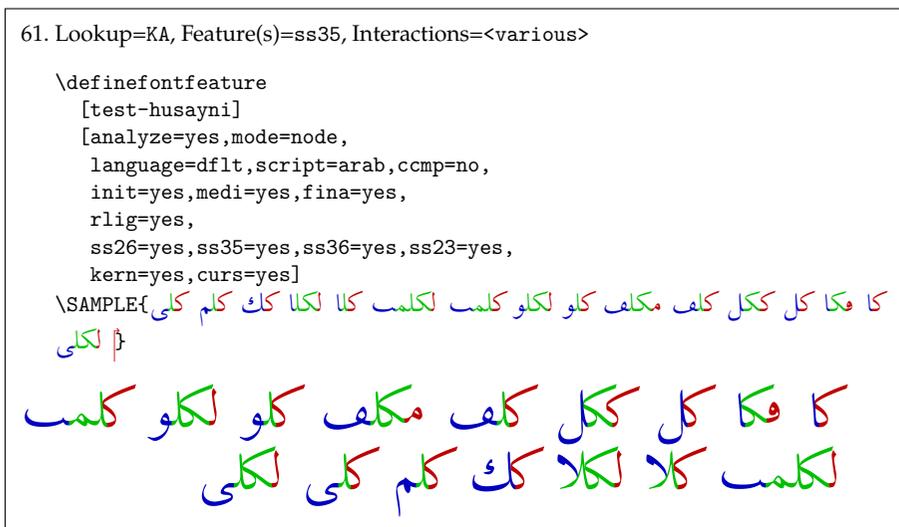


Figure 2: Example 61 of the GSUB lookup list.

about any other typesetting system in the world.<sup>3</sup> Note the use of alternate substitution for end-of-verse indicators. I can hardly overemphasize the huge progress that *fatihah.pdf* represents, although there is still much farther to go!

## Conclusion

At my projected presentation at the DANTE e.V. meeting in Dortmund 2010 I intend to describe the OpenType machinery in more detail, and give some workshops or tutorials for its use. It is hard to overemphasize the time commitment required by this project. It is a full time job, with frequent 12–14 hours a day, virtually seven days a week. My deepest thanks to DANTE e.V. for this year’s grant to support a portion of this project, and I look forward to my projected presentation.

<sup>3</sup>Thomas Milo’s DecoType provides another advanced approach to the problem of culturally authentic Arabic script, and can presently be accessed through a plugin to InDesign. Our system will be in the end, we believe, more flexible, and especially *readable* for most advanced typesetting tasks. Advanced calligraphy programs for Arabic-script also exist, but they are not designed for paragraph typesetting but rather for illustration purposes.



Figure 3: First page of Fatihah.

## Dynamisches Ausblenden von Text und Erstellung von Lückentexten

### Uwe Ziegenhagen

Anwendungen für das dynamische Ausblenden von Textteilen gibt es viele: Ein Lebenslauf soll mal mit, mal ohne Angabe von Abschlussnoten gesetzt werden, für die Schule sollen Aufgabenblätter mal mit, mal ohne Lösungsweg erstellt werden. Ebenfalls beliebt in der Schule ist das Erstellen von Lückentexten, in denen statt des Lösungswortes nur eine entsprechende Anzahl von Unterstrichen oder ähnlichen Zeichen findet. Mit L<sup>A</sup>T<sub>E</sub>X ist es recht einfach, solche Anforderungen umzusetzen. Bei der Gelegenheit lässt sich auch den Umgang mit Befehlsdefinitionen üben.

### Ausblenden über die Textfarbe

Über das Setzen der Textfarbe auf weiß lässt sich Text zwar optisch ausblenden – was für reine Druckdokumente zwar ausreichend sein mag, in der PDF-Datei lässt sich der ausgeblendete Text aber trotzdem markieren und in die Zwischenablage kopieren:

Hallo, ich bin ein Text, in dem das Wort	ausgeblendet wird.
--	--------------------

```

1 \documentclass{article}
2 \usepackage{xcolor}
3 \newcommand{\blende}[1]{\textcolor{white}{#1}}
4 \begin{document}
5
6 Hallo, ich bin ein Text, in dem das Wort \blende{Phantom} ausgeblendet wird.
7
8 \end{document}

```

### Text verstecken mit `\phantom{}`

Um auch in elektronischen Dokumenten Textteile wirksam auszublenden, stellt T<sub>E</sub>X den `\phantom` Befehl bereit. Im folgenden Beispiel wird das Makro `\blende` in einem ersten Ansatz doppelt definiert. Je nachdem, ob ausgeblendet werden soll oder nicht, wird entweder die eine oder die andere Definition genutzt.

Hallo, ich bin ein Text, in dem das Wort	ausgeblendet wird.
--	--------------------

```

1 \documentclass{article}
2 \newcommand{\blende}[1]{\phantom{#1}}
3 %\newcommand{\blende}[1]{#1}
4 \begin{document}
5
6 Hallo, ich bin ein Text, in dem das Wort \blende{Phantom} ausgeblendet wird.
7
8 \end{document}

```

Das manuelle Auskommentieren von Befehlen ist nicht sonderlich elegant, daher können wir mithilfe des `ifthen`-Pakets [2] ein entsprechendes Schalterkommando definieren, das über die Werte `true` oder `false` die Definition des richtigen Makros `\blende` übernimmt.

Die Syntax des Befehls lautet `\ifthenelse{Bedingung} {Wahr} {Falsch}`, `Bedingung` ist dabei ein boolescher Ausdruck. Im `{true-Teil}` stehen die Befehle, die ausgeführt werden, wenn dieser Ausdruck »wahr« ergibt, im `{false-Teil}` die Befehle, die bei »falsch« ausgeführt werden. Wenn die Variable `ausblenden` auf `true` gesetzt ist, wird `\blende` mit `\phantom` Aufruf definiert, ansonsten erfolgt nur die einfache Ausgabe des als Parameter übergebenen Textes.

Über eine Änderung der Befehlsdefinition können wir das Beispiel noch deutlich einfacher gestalten. Das Kommando `\blende{}` wird nur einmal definiert, in der Definition wird überprüft, ob der boolesche Ausdruck `ausblenden` auf `true` oder `false` gesetzt und der übergebene Text entweder mit oder ohne `\phantom` Aufruf ausgegeben wurde.

Hallo, ich bin ein Text, in dem das Wort	ausgeblendet wird.
--	--------------------

```

1 \documentclass{article}
2 \usepackage{ifthen}
3 \newboolean{ausblenden} \setboolean{ausblenden}{true}
4 \ifthenelse{\boolean{ausblenden}}
5   {\newcommand{\blende}[1]{\phantom{#1}}} % der TRUE-Teil
6   {\newcommand{\blende}[1]{#1}} % der FALSE-Teil
7 \begin{document}
8
9 Hallo, ich bin ein Text, in dem das Wort \blende{Phantom} ausgeblendet wird.
10
11 \end{document}

```

Hallo, ich bin ein Text, in dem das Wort	ausgeblendet wird.
--	--------------------

```

1 \documentclass{article}
2 \usepackage{ifthen}
3 \newboolean{ausblenden} \setboolean{ausblenden}{true}
4 \newcommand{\blende}[1]{%
5   \ifthenelse{\boolean{ausblenden}}{\phantom{#1}}{#1}}
6 \begin{document}
7
8 Hallo, ich bin ein Text, in dem das Wort \blende{Phantom} ausgeblendet wird.
9
10 \end{document}

```

## Das Paket ifthen

Abgesehen von der oben vorgestellten Nutzung des Paketes `ifthen` zur Steuerung von Befehlsdefinitionen kann es natürlich auch allein genutzt werden, um Textteile auszublenden. Das folgende Beispiel zeigt noch einmal die minimalen Einstellungen, um Text per `ifthen` auszublenden.

Hier steht der Text, der bei <code>greeting=true</code> gesetzt wird.
---

```

1 \documentclass[ ]{article}
2 \usepackage{ifthen}
3 \newboolean{greeting} \setboolean{greeting}{true}
4 \begin{document}
5
6 \ifthenelse{\boolean{greeting}}{Hier steht der Text, der bei greeting=true
7 gesetzt wird.}{Hier steht der Text, der bei greeting=false gesetzt wird}
8
9 \end{document}

```

## Erstellung von Lückentexten

Bei der Erstellung von Lückentexten wird normalerweise nicht nur der Text ausgeblendet, es wird auch eine visuelle Unterstützung in Form einer Linie oder einer Folge von Unterstrichen gegeben, die die Länge des einzutragenden Wortes angibt. Für die Erstellung von Lückentexten können wir die Befehlsdefinition etwas abändern. Basierend auf einem Codeschnipsel aus dem Internet [1] können wir unser Beispiel abändern: Für jedes Wort, das ausgeblendet werden soll, setzt dieser Befehl `\luecke{}` eine gepunktete Lücke von der 2,7-fachen Länge des übergebenen Textes. Der Verlängerungsfaktor von 2,7 ist hier nur ein Richtwert, der auch in Abhängigkeit von Schriftart, Zeilenabstand und ähnlichen Einflussfaktoren adjustiert werden sollte.



Die Befehlsdefinition des Lückentextes ist hinreichend einfach, um an diesem Beispiel in einer der nächsten Ausgaben der DTK die Erstellung eines kleinen L<sup>A</sup>T<sub>E</sub>X-Pakets zu zeigen.

Anmerkungen und Anregungen zu diesem Artikel sind jederzeit willkommen, die Beispiel habe ich auch auf <http://www.uweziegenhagen.de> ablegt.

## Literatur

- [1] Johannes Almer: *Automatischer Lückentext in L<sup>A</sup>T<sub>E</sub>X und LyX*; CTAN; 2009; [http://www.edu-maphy.de/joomla/index.php?option=com\\_content&view=article&id=72:automatischer-lueckentext-in-latex-und-lyx&catid=62:latexlyx-tipps-und-tricks&Itemid=61](http://www.edu-maphy.de/joomla/index.php?option=com_content&view=article&id=72:automatischer-lueckentext-in-latex-und-lyx&catid=62:latexlyx-tipps-und-tricks&Itemid=61).
- [2] David Carlisle: *The ifthen package*; Anleitung; 2001; <http://www.ctan.org/tex-archive/macros/latex/unpacked/>.

# L<sup>A</sup>T<sub>E</sub>X-Figuren mit Hilfe von OpenOffice.org 3 Draw

Herbert Möller

Das leistungsfähige, kostenlose Zeichenprogramm OpenOffice.org 3 Draw [7] (im Folgenden 00oDraw abgekürzt) und ein Perl-Filterprogramm werden benutzt, um auch komplizierte editierbare Figuren bequem mit der L<sup>A</sup>T<sub>E</sub>X-picture-Umgebung zu erzeugen. Das Filterprogramm 00opict.pl [6] verarbeitet von 00oDraw exportierte PostScript-Dateien und liefert gebrauchsfertigen L<sup>A</sup>T<sub>E</sub>X-Code. Da der PostScript-Text in den von 00oDraw unterstützen Betriebssystemen Linux, Mac OS, Unix und Windows übereinstimmt, kann 00opict.pl auf jeder dieser Plattformen verwendet werden. Günstig ist es auch, dass alle mit 00oDraw gezeichneten Kurven in der PostScript-Datei als kubische Bézier-Kurven dargestellt werden; denn das von 00opict.pl unterstützten L<sup>A</sup>T<sub>E</sub>X-Paket pict2e [2] gibt diese Kurven sehr platzsparend als Vektorgrafik wieder.

## Einführung

Vor drei Jahren wurde eine neue Methode zur Erzeugung von editierbaren Figuren mit der L<sup>A</sup>T<sub>E</sub>X-picture-Umgebung im Comprehensive T<sub>E</sub>X Archive Network (CTAN) [4] veröffentlicht. Sie heißt »GaPFilL« (*Graphics as Post-Script Filtered for L<sup>A</sup>T<sub>E</sub>X*), weil sie aus den folgenden vier Schritten besteht:

1. Erzeugung der gewünschten Figur mit einem Geometrie- oder Zeichenprogramm;
2. »Drucken« (oder Exportieren) der Konstruktion als PostScript-Datei;
3. Anwenden eines Filterprogramms auf den PostScript-Text;
4. Kopieren der benötigten Teile des Ausgabertextes in ein L<sup>A</sup>T<sub>E</sub>X-Dokument.

Da die Schritte 1. und 2. sowie die im L<sup>A</sup>T<sub>E</sub>X-Dokument verwendeten Pakete variieren, sind verschiedene Filterprogramme nötig. Die beiden 2006 in das CTAN gestellten Filterprogramme `CABebez.pl` und `CABpict.pl` wurden für die Macintosh-Version der Geometrie-Software *Cabri-géomètre*<sup>TM</sup> II entwickelt. Das erste setzt nur das Makropaket `ebezier` [1] voraus, das zweite benutzt außerdem das Paket `pict2e` von 2004.

Auf der Website *Mathkompass* [3] sind in der Abteilung »Programme« zwei Filter `CDTex` und `CDebez.pl` für das nicht mehr erhältliche Zeichenprogramm *ClarisDraw* vorhanden. Der erste Filter ist ein AppleScript-Programm, das 2002 geschrieben wurde, um eine größere Zahl von Figuren mit der Folientechnik von *ClarisDraw* durchzupausen. Das auf `CABebez.pl` basierende Programm `CDebez.pl` profitiert davon, dass *ClarisDraw* in dem eigenen PostScript-Übersetzer kubische Bézier-Kurven darstellt.

Die Dokumentationen in [4] und [5] beschreiben die `GaPFilL`-Methode anhand der beiden Prototypen `CABebez.pl` und `CABpict.pl`. In den nächsten beiden Abschnitten werden deshalb nur Eigenschaften von `OOoDraw` behandelt, die für die Erzeugung von L<sup>A</sup>T<sub>E</sub>X-Figuren nützlich sind. Dann folgt die Erklärung des Einsatzes von Perl mit Mac OS und Windows. Der letzte Abschnitt enthält einige Erläuterungen der Besonderheiten des Filterprogramms `OOopict.pl`, die von denen des Programms `CABpict.pl` verschieden sind.

## Das Zeichenprogramm

`OOoDraw` ist Teil des Office-Pakets `OpenOffice.org 3`, das kostenlos aus [7] heruntergeladen werden kann (424 MB). Verglichen mit *ClarisDraw* ist `OOoDraw` noch leistungsfähiger. Hier werden nur einige Hinweise wiedergegeben, die in der Online-Hilfe nicht leicht zu finden sind.

### Anpassung von Symbolleisten

Wird `OOoDraw` hauptsächlich zur Erzeugung von L<sup>A</sup>T<sub>E</sub>X-Figuren verwendet, so sollten im Untermenü »Symbolleisten« des Menüs »Ansicht« die Einträge »Linie und Füllung«, »Optionen«, »Standard«, »Zeichnen« und »Farbleiste« aktiviert werden. Der Indexeintrag »Anpassen; Symbolleisten« beschreibt, wie sich Schaltflächen in den Symbolleisten hinzufügen oder ändern lassen.

Für die Leiste »Zeichnen« sind folgende sichtbare Schaltflächen zu empfehlen: »Auswahl«, »Linie«, »Quadrat, ungefüllt«, »Rechteck, ungefüllt«, »Kreis, ungefüllt«, »Kreisbogen«, »Ellipse, ungefüllt«, »Ellipsenbogen«, »Maßlinie«, »Polygon«, »Punkte«, »Kurve«, »Bézier schließen«, »Punkte bearbeiten« und »Position und Größe«. Da das Filterprogramm keine gefüllten Objekte und keine Pfeile verarbeitet, sollten diese nicht in der Leiste »Zeichnen« wählbar sein.

## Zeichnen von Kurven

Das Werkzeug »Kurve« ist besonders nützlich, weil OOoDraw und das Paket `pict2e` es ermöglichen, beliebige Kurven als Vektorgrafik mit L<sup>A</sup>T<sub>E</sub>X zu erzeugen. Die Hilfefunktionen »Kurven; zeichnen« und »Kurven; bearbeiten« erläutern, wie diese kubischen Bézier-Kurven handzuhaben sind. Das Filterprogramm `OOopict.pl` erkennt Linien und Kurven an den Zeilenenden in dem PostScript-Text. *Deshalb darf in einer Kurvenkette keine Linie vorkommen.*

Da `pict2e` keine gepunkteten kubischen Bézier-Kurven liefert, approximiert `OOopict.pl` diese Kurven durch vier quadratische gepunktete Bézier-Kurven.

Aus hellblauen oder grünen Zwei-Strecken-Polygonen in OOoDraw erzeugt `OOopict.pl` quadratische Bézier-Kurven. Diese Kurven, die Parabeln darstellen, sind in den Folien von OOoDraw nicht sichtbar. Um sichtbare Parabeln zu erhalten, kann die Tatsache benutzt werden, dass jede quadratische Bézier-Kurve eine spezielle kubische Bézier-Kurve ist (siehe Abbildung 1).

Die durch das Polygon *AED* erzeugte quadratische Bézier-Kurve ist identisch mit der kubischen Bézier-Kurve zu den Kontrollpunkten *A*, *B*, *C* und *D*, wobei  $|BE| = \frac{1}{3} |AE|$  ist und *ADFB* ein Parallelogramm darstellt. Die Länge von Strecken kann mit dem Werkzeug »Maßlinie« von OOoDraw bestimmt werden.

Bei der folgenden Erläuterung des Schließens von Kurven oder Polygonen wird auch auf einige nützliche Zeichenhilfen hingewiesen. Doppelklicken in der Nähe des Startpunkts stellt die direkte Schließmethode dar. Zur Kontrolle kann man auf der linken Seite der »Statusleiste« am unteren Rand des Fensters beobachten, ob die Anzahl der Punkte unverändert bleibt. Ist es schwierig, Objektpunkte zu »fangen«, so kann man den »Fangbereich« im Menü »Raster« der »Einstellungen« von OOoDraw vergrößern oder mit dem Schieberegler auf der rechten Seite der Statusleiste den Zoomfaktor der Ansicht ändern.

Die einfachste Schließmethode erhält man mit dem Werkzeug »Bézier schließen«, das Kurven und Polygone durch eine Linie schließt. Wie oben erläutert

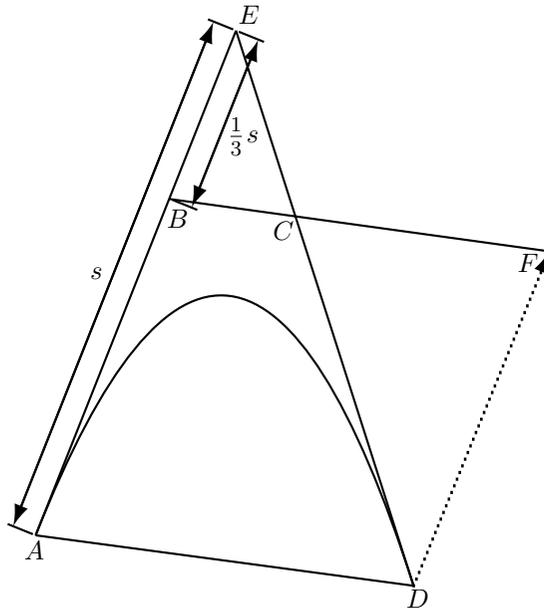


Abbildung 1: Visualisierung von quadratischen Bézier-Kurven (Parabeln)

wurde, ist diese Linie im Falle einer Kurve in ein Kurvensegment umzuwandeln. Das kann mit den Werkzeugen »Glatter Übergang« oder »Symmetrischer Übergang« aus der Symbolleiste »Punkte bearbeiten« geschehen.

### Ebenen und PostScript

Die meisten Zeichenprogramme stellen »Ebenen« für die Anordnung der Elemente einer Seite zur Verfügung. In jeder Ebene können für die Objekte die drei Zustände »sichtbar«, »druckbar« oder »gesperrt« ein- oder ausgeschaltet werden.

Die wichtigste Anwendung für die GaPFilL-Methode ist das Durchpausen von Figuren. 00oDraw stellt drei vordefinierte Ebenen zur Verfügung: Layout, Controls und Maßlinien. Die Layout-Ebene ist der voreingestellte Arbeitsbereich, in den auch der Inhalt einer Datei geladen wird, die man in 00oDraw öffnet. Da 00oDraw das JPEG-Format akzeptiert, können sogar Fotos durchgepaust werden, die mit einer Digitalkamera aufgenommen wurden.

Für das Durchpausen muss mindestens eine Ebene hinzugefügt werden, indem man »Ebene. . .« im Menü »Einfügen« wählt oder mit der rechten Maustaste den Ebenen-Registerbereich unterhalb des Zeichenfeldes anklickt. Zunächst sollte mit Farben durchgepaust werden, die sich gut vom Original abheben. Vor dem Speichern sind die im nächsten Abschnitt angegebenen Farben zu wählen, mit deren Hilfe Informationen an das Filterprogramm übermittelt werden.

Der PostScript-Übersetzer berücksichtigt die Inhalte aller Ebenen, auch wenn sie unsichtbar oder geschützt sind. Vor dem Exportieren in eine EPS-Datei mit »Exportieren. . .« im Menü »Datei« müssen deshalb die Inhalte aller nicht mehr benötigten Ebenen gelöscht werden. In dem Fenster »EPS Export Optionen«, das sich öffnet, wenn man »Sichern« wählt, sind folgende Einstellungen für `00opict.p1` notwendig: keine Vorschau, Level 2, Farbe und keine Kompression.

## Verwendung von Farben

Mit Hilfe von Farben werden Zusatzinformationen an das Filterprogramm übermittelt. `00oDraw` bietet mehrere Farblisten an. Im Folgenden benutzen wir zwölf der ersten vierzehn Farben der voreingestellten Liste »standard.soc'. Alle Farben der Liste werden in der »Farbleiste« unterhalb des Zeichenfeldes angezeigt. Bewegt man den Cursor über eines der farbigen Quadrate, so erscheint der Name der Farbe. Da die englischen Bezeichnungen derselben Farben aus der Liste »html.soc« gebräuchlicher sind, werden sie intern in `00opict.p1` verwendet. Diese Farbnamen stehen in der Tabelle 1 am Ende dieses Abschnitts in Klammern hinter den »Standardbezeichnungen« .

Es gibt drei Möglichkeiten, eine Linie oder Kurve einzufärben. Man kann eine Farbe im Voraus festlegen, indem man auf die entsprechende Fläche im Menü »Linienfarbe« der Symbolleiste »Linie und Füllung« klickt. Die Farbe einer ausgewählten Linie oder Kurve lässt sich durch Klicken mit der rechten Maustaste auf das Feld der neuen Farbe in der Farbleiste ändern. Schließlich wird durch Ziehen einer Farbe der Farbleiste auf eine beliebige Linie oder Kurve diese mit der ausgewählten Farbe gefärbt.

Im Folgenden kann ein »Objekt« eine Linie, ein Polygon, ein Kreis, ein Kreisbogen, eine Ellipse, ein Ellipsenbogen oder eine kubische Bézier-Kurve sein.

- *Durchgezogene Objekte* werden mit der voreingestellten Farbe *schwarz* gezeichnet.
- *Gepunktete Objekte* erhält man mit der Farbe *hellgrün*.

- Pfeile mit durchgezogenem Schaft müssen als türkis-farbene Linien eingegeben werden. Blaue Linien erzeugen Pfeile mit gepunktetem Schaft.
- Durchgezogene quadratische Bézier-Kurven ergeben sich aus hellblauen Polygonen mit zwei Strecken. Dieselben Polygone in grün liefern gepunktete quadratische Bézier-Kurven.
- Text- oder Formelpositionen werden durch Linien mit der Farbe helltürkis markiert. Damit sich die »bounding box« korrekt berechnen lässt, sollte die markierende Strecke ungefähr die links unten beginnende Diagonale des kleinsten Rechtecks sein, das den Text beziehungsweise die Formel umschließt. An der Position des Anfangspunkts der Strecke erscheint in der L<sup>A</sup>T<sub>E</sub>X-Ausgabe eine fortlaufende Nummer.
- Die übrigen fünf Farben werden zum Füllen, Schraffieren oder Punktieren von Flächen benutzt, die von geschlossenen Polygonen berandet sind. Das Filterprogramm zerlegt dazu die von einem Polygon gebildete Fläche in Dreiecke mit dem Ausgangspunkt des Polygons als gemeinsamem Eckpunkt. Um Überschneidungen zu vermeiden, ist also die gegebene Fläche durch Polygone so zu unterteilen, dass für jedes Polygon beim Verbinden des Ausgangspunkts mit den übrigen Eckpunkten jeweils Dreiecke entstehen, die höchstens eine Seite mit jedem der anderen Dreiecke gemeinsam haben (siehe [4] oder [5]).  
Hellrote Polygone werden ohne Rand gefüllt. Zum Schraffieren mit sichtbaren Randstrecken ist die Polygonfarbe rot zu wählen, ohne Randstrecken die Farbe braun. Hat das Polygon die Farbe magenta, so wird die Fläche punktiert und die Randstrecken sind sichtbar; mit der Farbe hellmagenta lässt sich die Fläche ohne Rand punktieren. Die bisherigen Farbuordnungen sind in Tabelle 1 zusammengefasst.
- Alle Zeichenelemente, die andere Farben haben, können als Zeichenhilfen dienen, weil sie von 00opict.pl ignoriert werden.

## Anwendung des Perl-Systems

Für den Einsatz des Filterprogramms 00opict.pl benötigt man das Perlsystem und einen Texteditor, der es ermöglicht, ein Skript oder eine Kommandozeile auszuführen, während eine Textdatei im Frontfenster geöffnet ist.

### Mac OS X

Das Betriebssystem stellt das einsatzbereite Perlsystem zur Verfügung. Als geeigneter Texteditor ist TextWrangler 2.3 kostenlos erhältlich (<http://www.barebones.com/products/textwrangler>). Es ist ein Vielzweckeditor, der als Nebenprodukt des professionellen HTML- und Texteditors BBEdit<sup>TM</sup> 9 von

Tabelle 1: Farbzunordnungen

<i>Typ</i>	<i>Farbe</i>
durchgezogenes Objekt	schwarz (black)
gepunktetes Objekt	hellgrün (lime)
durchgezogener Pfeil	türkis (teal)
gepunkteter Pfeil	blau (navy)
durchgez. quad. Bézier-Kurve	hellblau (blue)
gepunkt. quad. Bézier-Kurve	grün (green)
Text- oder Formelmarker	helltürkis (cyan)
ohne Rand füllen	hellrot (red)
mit Rand schraffieren	rot (maroon)
ohne Rand schraffieren	braun (olive)
mit Rand punktieren	magenta (purple)
ohne Rand punktieren	hellmagenta (magenta)

Bare Bones Software, Inc. entwickelt wurde. **TextWrangler** nutzt das Perlssystem direkt als Unix-Skriptumgebung. Deshalb wird darauf über das Menü »#!« zugegriffen, das seinen »Namen« von dem Anfang der ersten Zeile jedes Unix-Skripts hat.

**00opict.pl** gehört in den Unterordner »Unix Filters«, der durch den Befehl »Open Filters Folder« im Untermenü »Unix Filters« des Menüs »#!« geöffnet werden kann. Befindet sich ein **00oDraw-PostScript-Text** im Frontfenster, so braucht man nur **00opict.pl** im Untermenü »Unix Filters« zu wählen. Im Bruchteil einer Sekunde wird dann der PostScript-Text durch den entsprechenden L<sup>A</sup>T<sub>E</sub>X-Code ersetzt.

### Windows und andere Betriebssysteme

Das vollständige, installationsfertige Perlssystem **ActivePerl** ist für Windows, Linux, Mac OS X, Solaris, AIX und HP-UX kostenlos von ActiveState Software Inc. erhältlich (<http://www.activestate.com/activeperl>). Diese Firma bietet auch den kostenlosen Texteditor **Komodo Edit 5.1** an ([http://www.activestate.com/komodo\\_edit](http://www.activestate.com/komodo_edit)), der ein Nebenprodukt der professionellen **Komodo<sup>TM</sup> IDE** ist.

Nach der Installation von **ActivePerl** kann **00opict.pl** zum Beispiel in dem Ordner »C:\Perl\eg\« gespeichert werden. Hat man zum ersten Mal einen

OOoDraw-PostScript-Text in Komodo Edit geladen und mit »ctrl-A« ausgewählt, so kann man »Run Command« im Menü »Tools« anklicken. In die Kommandozeile des sich öffnenden Fensters wird »C:\Perl\eg\00opict.pl« (oder der Pfad des Ordners mit 00opict.pl) eingegeben. Vor der Ausführung der Kommandozeile ist das Eingabefeld »Pass selection as input« zu aktivieren. Durch Aktivierung des dritten Eingabefeldes wird die Kommandozeile in der Werkzeugbox gespeichert. Dort kann sie durch Doppelklicken auf ihren Namen ausgeführt werden.

## Das Filterprogramm 00opict.pl

Das Perl-Filterprogramm 00opict.pl ist 701 Zeilen lang – ohne Kommentarzeilen, die mit »#« beginnen und ohne die Zeilen mit schließenden Klammern »}«. Die Zeilennummern der folgenden Ausschnitte gehören nicht zum Programm.

```

1 #!/usr/bin/perl -w
2 # 00opict.pl
3 # (c) Copyright 2009 H. Moeller (mollerh@math.uni-muenster.de).
4 # Version 1.3 for OpenOffice.org 3 Draw with Perl 5.8, and with the
   LaTeX-package 'pict2e' from 2008 or with 'pict2e' (2004) and 'ebezier'.
[ ... ]
14 # If the package pict2e is from 2008 or later (else $pictnew=0):
15 $pictnew = 1;

```

Da es zwei Versionen von `pict2e` gibt, dient die Flagge `$pictnew` in Zeile 15 dazu, von der `CABpict.pl`-Umgebung auf die neue umzuschalten, die nicht mehr das Paket `ebezier` benötigt, weil die 2008-Version von `pict2e` gepunktete quadratische Bézier-Kurven bietet. Außerdem ist der Betrag der Schranken für die Liniensteigungskomponenten von 1000 auf 16383 erhöht, was in `00opict.pl` auch durch `$pictnew` umgeschaltet wird.

Normalerweise befindet sich die Datei `pict2e.sty` in einem Unterordner `\dots\tex\latex\pict2e/`. Wenn man die Datei nicht findet, um das Versionsdatum nachzusehen, kann man einen Test mit einem L<sup>A</sup>T<sub>E</sub>X-Programm durchführen, das `\usepackage[pdftex]{pict2e}` in der Präambel und eine Zeile `\put(0,0){\line(1001,1000){10}}` im Dokument enthält. Bei der alten Version wird ein Fehler gemeldet. Ist bekannt, wo sich `pict2e.sty` befindet, so kann die Datei ersetzt werden, indem man den Ordner `pict2e` aus [2] herunterlädt, `pict2e.ins` mit L<sup>A</sup>T<sub>E</sub>X ausführt und die entstehende `.sty`-Datei für den Austausch verwendet.

```

46 @lines = <>;
47 do {
48   $_ = $lines[$i++];

```

```

49  if ((/ l /o) or (/ ct /o) or (/ p[cs ][\se]/o)) {
50    s/\d+ lw \d+ lj //go;
51    s/ m //go;
52    if (/ c /o) {
53      s/(\d\.\d+) (\d\.\d+) (\d\.\d+) c //o;
54      $c=$1.$2.$3. " ";
55    }
56    $t="";
57    $l="";
58    $tflag=0;
59    $fflag=0;
60    while ($_ !~ (/p[cs ][\se]/o)) {
[ ... ]
88    if ($tflag and $fflag) {
89      $curve[++$#curve] = $c.$t;
90    }
91    elsif ($fflag) {
92      $polygon[++$#polygon] = $c.$l;
93    }
94  }
95 }
96 until $i == $#lines;

```

Beim Filtern der Textzeilen für L<sup>A</sup>T<sub>E</sub>X treten drei Schwierigkeiten auf. 1) Wenn aufeinanderfolgende Elemente (Linien, Polygone, Kurven) dieselbe Farbe haben, stehen die Farbdaten nur beim ersten Element (Zeilen 52 bis 54). 2) Lange Datenzeilen werden von dem PostScript-Übersetzer umbrochen, bevor die charakterisierenden Endzeichen (»ps«, »pc« oder »ef«) erreicht sind. Deshalb müssen die aufeinanderfolgenden Zeilen in Abhängigkeit von den Datentrennern »l« (»lineto«) bei Linien und »ct« (»curveto«) bei Kurven in den Zeichenketten »\$l« und »\$t« zusammengefasst werden. 3) Ob ein Objekt zu füllen ist, kann nur an dem Zeilenende »ef« erkannt werden. Hier wird die Flagge »\$fflag« verwendet, um zu entscheiden, welche Zeichenketten in den Arrays »@curve« oder »@polygon« zu speichern sind (Zeilen 88 bis 92).

Da 00opict.pl nur zwei Arrays für die Speicherung der Datenketten benötigt, reichen 130 Zeilen für die Aufrufe der passenden Unterprozeduren. Rund 120 Zeilen und die Unterprozedur »arc« konnten eliminiert werden, weil der PostScript-Übersetzer von 00oDraw alle Kurven als kubische Bézier-Kurven darstellt.

Einige der übrigen acht Unterprozeduren unterscheiden sich von denen in CABpict.pl. In der Unterprozedur »best« ist jetzt berücksichtigt, dass  $2^{31} - 1$  die größte ganze Zahl für Perl darstellt. Außerdem wurde ein Eingang für die obere Schranke der approximierenden Zähler und Nenner eingeführt.

In der Unterprozedur »lin« werden gepunktete Linien nun durch den Befehl »\multiput« von L<sup>A</sup>T<sub>E</sub>X erzeugt. In Abhängigkeit von der Flagge \$pictnew gibt es zwei Möglichkeiten, gepunktete quadratische Bézier-Kurven mit der Unterprozedur »qbez« und gepunktete kubische Bézier-Kurven mit »cbez« zu zeichnen. Wie oben erwähnt wurde, erreicht 00opict.pl das letztere, indem es die kubische Bézier-Kurve durch vier quadratische Bézier-Kurven approximiert. Das Ergebnis ist jedoch nur befriedigend, wenn die kubische Bézier-Kurve keine Wendepunkte oder Selbstschnittpunkte enthält. Das lässt sich aber immer erreichen, indem die kubische Bézier-Kurve mit 00oDraw in passende Segmente zerlegt wird.

## Literatur

- [1] Gerhard A. Bachmaier: *Using ebezier*; 2002; Paket in CTAN:/macros/latex/contrib/ebezier.
- [2] Hubert Gäßlein, Rolf Niepraschk and Josef Tkadlec: *The pict2e-package*; 2008; Paket in CTAN:/macros/latex/contrib/pict2e.
- [3] Herbert Möller: *Mathkompass*; <http://www.math.uni-muenster.de/u/mollerh>.
- [4] Herbert Möller: *How to generate L<sup>A</sup>T<sub>E</sub>X picture environments using the GaPFilL method*; 2006; CTAN:/graphics/gapfill/GaPDoc1.pdf.
- [5] Herbert Möller: *Die GaPFilL-Methode zur Erzeugung von L<sup>A</sup>T<sub>E</sub>X-picture-Umgebungen*; 2006; Die T<sub>E</sub>Xnische Komödie, 1/2006, S. 25–43.
- [6] Herbert Möller: *How to generate L<sup>A</sup>T<sub>E</sub>X picture environments using OpenOffice.org 3 Draw*; 2009; CTAN:/graphics/gapfill/GaPDoc2.pdf.
- [7] OpenOffice.org: *OpenOffice.org 3*; 2009; <http://www.openoffice.org>.

## Von `pageref` zu `hyperpage`

Herbert Möller

Bei dem Einsatz des Pakets `hyperref` [1] mit `pdfTeX` [2] ergeben die aus `\pageref` erzeugten Links oft fehlerhafte Sprünge. Dieses Problem und systematische Korrekturen bei der Indexerstellung werden mit Perl-Filterprogrammen gelöst.

### Einleitung

Da die Mathematik eine deduktive Wissenschaft ist, finden sich bei Beweisen in Lehrbüchern oft Formulierungen der folgenden Art:

»Nach §3, Satz 3, gibt es ...«.

Dieser Hinweis steht (in einem sehr verbreiteten Buch) auf Seite 44, §3 reicht von Seite 17 bis Seite 26, und nach einigem Blättern findet man Satz 3 auf Seite 25. Welch einen Fortschritt brachte in dieser Situation die flexible Seitenangabe durch den `\pageref`-Befehl von `LATEX`! Noch viel bequemer sind Sprünge in einer mit `LATEX` erzeugten Datei auf einem Computerbildschirm, wenn das Paket `hyperref` verwendet wird. Mit einem einzigen Mausclick kommt man aus dem Inhaltsverzeichnis, dem Index, einem Symbolverzeichnis, einer Lesezeichenleiste oder von einer Textmarkierung zu der gewünschten Stelle und mit einem zweiten Klick wieder zurück.

Von dem `HyperTeX`-Projekt ausgehend schuf Sebastian Rahtz von 1995 bis 2001 den Kern von `hyperref`. Seit 2001 wird das Paket von Heiko Oberdiek betreut. Mit mehr als 220 `ChangeLog`-Einträgen und zwölf eigenen Paketen [3] zur Unterstützung von `hyperref` ist seine Leistung besonders dankens- und lobenswert.

In der oben beschriebenen Situation der Rücksprünge kommt man überraschenderweise beim Anklicken einer mit `\pageref` erzeugten Seitenzahl meistens auf der Seite an, die den Anfang des Abschnitts mit der zugehörigen `\label`-Einsetzung enthält. Nach Auskunft von Heiko Oberdiek handelt es sich dabei um ein grundsätzliches Problem, das nicht mit `hyperref` für `pdfTeX` zu beheben ist. Es hängt mit den symbolischen Adressen von `TEX` zusammen, die das `Portable-document-Format` nicht ohne Weiteres zur Erzeugung von »Ankern« für Hypertext verwenden kann.

Da der Fehler nicht auftritt, wenn `\pageref` und die symbolische Adresse durch `hyperpage` mit der zugehörigen Seitenzahl ersetzt wird, liegt es nahe, das Problem mit einem Filterprogramm zu lösen.

## Das Perl-Filterprogramm `Aux.pl`

Enthält eine `.tex`- oder `.ltx`-Datei eine `\label`-Einsetzung, so gibt es in der zugehörigen `.aux`-Datei eine Zeile mit einem charakteristischen Anfang und mit den entsprechenden Daten, zum Beispiel

```
\newlabel{sy:ganz}{{2.1}{17} ...}.
```

Dabei ist `sy:ganz` die Label-Bezeichnung, 2.1 die Abschnittsnummer und 17 die Zahl der Seite, auf der `\label{sy:ganz}` steht.

Das folgende Perl-Filterprogramm `Aux.pl` ersetzt in jeder `\newlabel`-Zeile der `.aux`-Datei `\newlabel` durch `s\pageref\` und die Abschnittskennzeichnung mit vier umgebenden Klammern durch `hyperpage`. Mit `\` und `/` werden die Sonderbedeutungen von `\` und `/` in Perl aufgehoben. Die dritte `s`-Zeile löscht alles, was nach der Seitenzahl bis zum Zeilenende folgt.

```
#!/usr/bin/perl -w
# Aux.pl
@lines = <>;
$pert = "#!/usr/bin/perl -w\n# Pagex.pl\n undef \$/;\n \$_ = <>\n";
do {
    $_ = $lines[$i++];
    if (/newlabel/) {
        s/\newlabel/s\pageref\/o;
        s/\}\{\{d+\.\?d*\}\}/hyperpage\/o;
        s/\}.\+$/\go;/o;
        $pert .= $_;
    }
}
until ($i == $#lines);
$pert .= "print STDOUT;\n";
print $pert;
```

Mit den extrahierten Zeilen wird ein neues Filterprogramm aufgebaut, das dann in der zugehörigen `.tex`- oder `.ltx`-Datei die `\pageref`-Ersetzungen durchführt. Wie Perl-Filterprogramme anzuwenden sind, steht in [4].

Die `.aux`-Datei mit der oben angegebenen `\newlabel`-Zeile wird durch das folgende Filterprogramm `Pagex.pl` ersetzt, wobei »x« eine einzufügende nähere Bezeichnung bedeutet. Falls das Projekt aus mehreren `TEX`-Dateien besteht, ist es bei der hohen Taktfrequenz der heutigen Computer sinnvoll, alle Ersetzungszeilen in einem Filterprogramm zusammenzufassen.

```
#!/usr/bin/perl -w
# Pagex.pl
  undef $/;
  $_ = <>;
[... ]
s/pageref\{sy:ganz/hyperpage\{17/go;
[... ]
print STDOUT;
```

Um mit `\hyperpage` korrekte Sprünge zu erreichen und im Index die richtigen Seitenzahlen zu erhalten, ist es zu empfehlen, durchgängig arabische Seitenzahlen zu verwenden, die mit den im PDF-Reader angezeigten physischen Seitenzahlen übereinstimmen. Dazu ist es notwendig, nach `\mainmatter` mit `\setcounter{page}{3}` die Seitenzahl anzupassen, wenn der Vorspann wie üblich die Titel- und die Impressumseite enthält.

Das Mathematiklehrbuch »*Elementare Zahlentheorie und Problemlösen*«, das kostenlos aus der Abteilung »*Zahlentheorie*« von [5] heruntergeladen werden kann, profitiert sehr wesentlich von `hyperref` mit der obigen Korrektur. Anstelle einer Satznummerierung haben alle Sätze suggestive Namen. Vor allem aus den Beweisen, in denen sie verwendet werden, und aus einem Satzverzeichnis mit mehr als 100 Einträgen sind sie durch Anklicken der hinzugefügten Seitenzahl erreichbar. Effizient ist auch ein Symbolverzeichnis mit 99 `\hyperpage`-Seitenangaben.

## Index-Filter

Falls die Pakete `ngerman` und `makeidx` benutzt werden und kein Formatänderungsdatei zum Ersetzen des Markierungszeichens " von `MakeIndex` vorliegt, kann man auf die `.idx`-Datei das folgende Perl-Filterprogramm anwenden, um die korrekte Behandlung von Umlauten durch `MakeIndex` zu erreichen.

```
#!/usr/bin/perl -w
# Index.pl
  undef $/;
  $_ = <>;
s/"s/{\ss}/go;
s"/"/go;
s/"/"/go;
s/active.dq...dq.prtct/"/go;
s/s {{/go;
print STDOUT;
```

Wird `hyperref` mit der voreingestellten Option `hyperindex=true` benutzt, so sind die Seitenzahlen nach Anwendung von `MakeIndex` in `\hyperpage`-Aufrufe

eingeschlossen. Wenn ein Indexeintrag mit einem Umlaut oder einem mathematischen Zeichen beginnt, muss die entsprechende Zeile per Hand einsortiert werden. Da das Umordnen wegen der vielen `\hyperpage`-Einträge mühsam wird, ist es günstiger, durch die Option `hyperindex=false` mit `MakeIndex` eine `.ind`-Datei zu erzeugen, die neben den Indexeinträgen nur die Seitenzahlen enthält. Nach Abschluss aller Korrekturen können dann mit dem folgenden Perl-Filterprogramm `Hyperind.pl` die `\hyperpage`-Anweisungen eingefügt werden.

```
#!/usr/bin/perl -w
# Hyperind.pl
undef $/;
$_ = <>;
s/\s(\d+)/ \\\hyperpage\{$1\}/go;
print STDOUT;
```

In der README-Datei von [1] werden hierfür unter »Hints« weitere Möglichkeiten angegeben.

## Literatur

- [1] Sebastian Rahtz und Heiko Oberdiek: *Hypertext marks in L<sup>A</sup>T<sub>E</sub>X*; 2009; Paket in CTAN:/macros/latex/contrib/hyperref.
- [2] Hàn Thê Thàn u. a.: *pdfT<sub>E</sub>X*; 2008; CTAN:/system/pdftex.
- [3] Heiko Oberdiek: *Bundle of packages*; 2009; Pakete in CTAN:/macros/latex/contrib/oberdiek.
- [4] Herbert Möller: *L<sup>A</sup>T<sub>E</sub>X-Figuren mit Hilfe von OpenOffice.org 3 Draw*; 2009; Die T<sub>E</sub>Xnische Komödie; 4/2009.
- [5] Herbert Möller: *Mathkompass*; <http://wwwmath.uni-muenster.de/u/mollerh>.

## (Kreuzwort-)rätsel setzen mit $\text{\LaTeX}$

Adelheid Grob

### Einleitung

Um mit  $\text{\LaTeX}$  Kreuzwörterrätsel zu setzen, stehen zwei Pakete zur Verfügung. Zum einen das ältere **crossword** von B. Kelly aus dem Jahr 1996, zum anderen **cpuzzle** von Gerd Neugebauer in der aktuellen Version vom September 2009. Mit dem ersteren können lediglich quadratische Kreuzwörterrätsel erzeugt werden. Das Paket von Gerd Neugebauer erlaubt hingegen das Setzen von Rätseln aller Art, seien es Standardkreuzwörterrätsel, Zahlenrätsel, Ausfüllrätsel oder seit neuestem auch Sudoku und Kakuro. Bei beiden Paketen ist eine Ausgabe der Lösung problemlos möglich. In diesem Artikel werden **crossword** und **cpuzzle** mit ihren Möglichkeiten vorgestellt.

### Das Paket **crossword**

Das Paket wurde 1996 von B. Hamilton Kelly geschrieben, um Kreuzwörterrätsel in Zeitschriften setzen zu können.

### Befehlsübersicht

Mit `\usepackage{crossword}` wird das Paket eingebunden. Es stellt die Umgebungen **crossword** und **crossword\*** zur Verfügung. Mit **crossword** wird das Rätsel gesetzt, mit **crossword\*** die Lösung ausgegeben.

Die Umgebung **crossword** besitzt zwei Argumente. Das erste enthält eine Ziffer, die die Anzahl der Spalten bzw. Zeilen angibt. Mit dem zweiten Argument wird mittels *Y* (für *Yes*) bzw. *N* (für *No*) gesteuert, ob die Lösungsbuchstaben mit ausgegeben werden sollen. Ersteres kann für das Entwerfen des Kreuzwörterrätsels hilfreich sein. In der Umgebung **crossword\*** dient das erste Argument ebenfalls zur Angabe der Spaltenanzahl, das zweite enthält einen Text, der über der Lösung des Rätsels stehen soll, beispielsweise »Lösung des Rätsels aus Heft 03/09«. In der Sternvariante sind zum einen die Kästchen kleiner, zum anderen fehlen die Positierungsnummern der Lösungswörter (siehe Beispiel 1).

## Beispiel 1: crossword und crossword\*

1			
2			

**ACROSS**

1 Baumteil (3)

2 Vorname (3)

**DOWN**

1 Weise (3)

```
\begin{crossword}{3}{N}
\clue{1}{A}{1}{1}{AST}{Baumteil}{3}
\clue{1}{D}{1}{1}{ART}{Weise}{3}
\clue{2}{A}{1}{3}{TED}{Vorname}{3}
\end{crossword}
```

**Lösung**

A	S	T
R		
T	E	D

```
\begin{crossword*}{3}{Lösung}
\clue{1}{A}{1}{1}{AST}{Baumteil}{3}
\clue{1}{D}{1}{1}{ART}{Weise}{3}
\clue{2}{A}{1}{3}{TED}{Vorname}{3}
\end{crossword*}
```

In Beispiel 1 wird der Befehl `\clue{1}{2}{3}{4}{5}{6}{7}` verwendet, mit dem die zu lösenden Aufgaben angegeben werden. Er besitzt sieben Argumente, die folgende Bedeutung haben:

*Nr. Bedeutung*

- 1 Positionierungsnummer
- 2 waagrecht oder senkrecht (*Across* oder *Down*)
- 3 x-Koordinate des ersten Buchstabens des Lösungswortes im Kreuzgitter; das oberste Kästchen hat die Koordinaten  $(x,y)=(1,1)$
- 4 y-Koordinate des ersten Buchstaben des Lösungswortes im Kreuzgitter
- 5 Lösung
- 6 Aufgabenstellung – falls  $\LaTeX$ -Makros verwendet werden, müssen diese mit einem vorangehenden `\noexpand` geschützt werden, z. B. `\noexpand \%`
- 7 Hinweis, wird in ( ) gesetzt (z. B. die Anzahl der Buchstaben des Lösungswortes)

## Probleme

Da die Standardsprache des Paketes Englisch ist, lauten die Überschriften der Aufgaben *Across* bzw. *Down*. Ein weiteres Problem ist, dass es nicht möglich ist, auf die Angabe eines Hinweises zu verzichten; lässt man das siebte Argument leer, werden dennoch die umschließenden Klammern gesetzt.

Die Aufgaben werden mit der `minipage`-Umgebung gesetzt, deren Breite vom Paketautor auf 70 mm festgesetzt wurde. Diese feste Breite stellt sich als problematisch heraus, wenn man ein Rätsel auf kleineren Formaten setzen möchte als A4. Dieses Problem lässt sich allerdings mit einer Anpassung auf `0.x\textwidth` leicht lösen, evtl. kann der Wert für `x` auch über eine neu definierte Option mitgegeben werden.

## Das Paket `cwpuzzle`

Wesentlich umfangreicher ist das Paket `cwpuzzle` von Gerd Neugebauer. Anders als im Paket `crossword` erfolgt das Setzen des Rätsels und der Aufgaben in zwei getrennten Umgebungen. Eine für alle Rätseltypen gleiche, sehr anschauliche Syntax erleichtert das Setzen der Rätsel darüber hinaus.

### Einbinden und Optionen

Eingebunden wird das Paket mittels `\usepackage[Optionen]{cwpuzzle}`, als `[Optionen]` stehen hierbei die in der nachfolgenden Tabelle genannten Möglichkeiten zur Verfügung:

<i>Option</i>	<i>Bedeutung</i>	<i>Standard</i>
<code>numbered</code>	die Platzierungsnummern der Antworten innerhalb des Rätsels (kleine Zahlen innerhalb der Kästchen) bleiben bei der Ausgabe der Lösung erhalten	keine Ausgabe der Nummern bei der Lösung
<code>nocenter</code>	das Rätsel wird nicht zentriert gesetzt	zentriertes Setzen des Rätsels innerhalb eines Absatzes
<code>unboxed</code>	die Aufgaben werden nicht zentriert, sondern als fortlaufender Text gesetzt	Setzen der Aufgaben innerhalb einer zentrierten <code>minipage</code> -Umgebung
<code>normalsize/</code> <code>small/large</code>	Größenangabe des Rätsels	<code>normalsize</code>
<code>german/</code> <code>ngerman</code>	Sprache der automatisch gesetzten Texte (z. B. «Wörter der Länge«)	Englisch

## Befehlsübersicht

## Allgemein

Es werden die Umgebungen `Puzzle{1}{2}` und `PuzzleClues{1}` zur Verfügung gestellt. Weiterhin werden für Zahlenrätsel die Umgebung `PuzzleWords{1}`, für Sudoku `Sudoku` und für Kakuro `Kakuro` definiert.

Der generelle Aufbau für das Setzen eines Kreuzwörtertsels wird in Beispiel 2 gezeigt. Die Umgebung `Puzzle{1}{2}` besitzt zwei Argumente – das erste gibt die Anzahl der Spalten (x), das zweite die Anzahl der Zeilen (y) an.

**Beispiel 2:** Die `Puzzle{1}{2}`-Umgebung

1	2	3		4	
5				6	7
8			9		
10					

```
\begin{Puzzle}{6}{4}
|[1] S|[2]C|[3]H| O|[4]N|* |.
|[5] A| H| A|* |[6]U|[7]N|.
|[8] F| U| S|[9]S|* | U|.
|[10]T| R| E| S| E| N|.
\end{Puzzle}
```

Man erkennt, dass der senkrechte Strich (|) als aktives Zeichen eingeführt wurde, dieser trennt die einzelnen Zellen des Kreuzwörtertsels voneinander. Ihm können verschiedene Zeichen bzw. Argumenten folgen:

Zeichen	Bedeutung
*	schwarzes Kästchen
[1]S	Positionierungsnummer des Lösungswortes im Kreuzwörtertsel, gefolgt von dem Buchstaben der entsprechenden Zelle
{ }	Kästchen ohne Rahmen (siehe Beispiel 12)
.	Zeilenende

Für die Angabe der Aufgaben steht die Umgebung `PuzzleClues{1}` zur Verfügung. Diese verfügt über ein Argument, das den Text angibt, der den Aufgaben vorangestellt werden soll, z. B. *Waagrecht*. Innerhalb dieser Umgebung ist der Befehl `\Clue{1}{2}{3}` definiert, dessen drei Argumente die folgende Bedeutung haben: Das erste gibt die Positionierungsnummer der Aufgabe im Kreuzwörtertsel an, das zweite enthält das Lösungswort und das dritte den Text der Aufgabe (siehe Beispiel 3).

### Beispiel 3: Die PuzzleClues{1}-Umgebung

#### Waagrecht

1 bereits 5  
 schwed. Gruppe  
 6 Vorsilbe 8  
 Körperteil 10  
 Teil einer Bar

#### Senkrecht

1  
 Getränk 2  
 Stadt in der  
 Schweiz 3 Tier  
 4 Im ... 7 Jetzt  
 9 Abk. für Som-  
 mersemester

```
\begin{PuzzleClues}{\textbf{Waagrecht}}
\Clue{1}{SCHON}{bereits}
\Clue{5}{AHA}{schwed. Gruppe}
\Clue{6}{UN}{Vorsilbe}\Clue{8}{FUSS}{↔
Körperteil}
\Clue{10}{TRESSEN}{Teil einer Bar}
\end{PuzzleClues}\[\[5pt]
\begin{PuzzleClues}{\textbf{Senkrecht}}
\Clue{1}{SAFT}{Getränk}
\Clue{2}{CHUR}{Stadt in der Schweiz}
\Clue{3}{HASE}{Tier} \Clue{4}{NU}{Im \dots}
\Clue{7}{NUN}{Jetzt}
\Clue{9}{SS}{Abk. für Sommersemester}
\end{PuzzleClues}
```

Für die Ausgabe der Lösung setzt man vor die `Puzzle`-Umgebung den Befehl `\PuzzleSolution[1]`; diese hat ein optionales Argument, mit dem gesteuert wird, ob die Positionierungsnummern mit ausgegeben werden sollen. Standardmäßig ist dieses auf `false` gesetzt. Die gestellten Aufgaben werden bei der Lösungsausgabe ignoriert (siehe Beispiel 4).

Um ein später im Dokument gesetztes Rätsel ungelöst ausgeben zu lassen, ist es nötig, dort den Befehl `\PuzzleUnsolved` vor der `Puzzle`-Umgebung aufzurufen.

### Beispiel 4: Ausgabe der Lösung

1	S	2	C	3	H	O	4	N	
5	A	H	A		6	U	7	N	
8	F	U	S	9	S			U	
10	T	R	E	S	E	N			

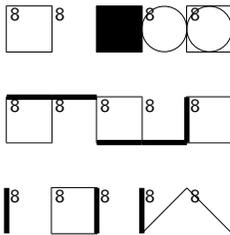
S	C	H	O	N	
A	H	A		U	N
F	U	S	S		U
T	R	E	S	E	N

```
\PuzzleSolution[true]
\begin{Puzzle}{6}{4}
|[1] S|[2]C|[3]H| O|[4]N|* |.
|[5] A| H| A| *|[6]U|[7]N|.
|[8] F| U| S|[9]S| *| U|.
|[10]T| R| E| S| E| N|.
\end{Puzzle}
```

```
\PuzzleSolution[false]
\begin{Puzzle}{6}{4}
|[1] S|[2]C|[3]H| O|[4]N|* |.
|[5] A| H| A| *|[6]U|[7]N|.
|[8] F| U| S|[9]S| *| U|.
|[10]T| R| E| S| E| N|.
\end{Puzzle}
```

## Verändern von Zellen

Um z. B. amerikanische Kreuzwortsrätsel<sup>1</sup> setzen zu können oder um die einzelnen Buchstaben eines Lösungswortes innerhalb des Kreuzwortsrätsels hervorheben zu können, stehen vordefinierte Zellen zur Verfügung. Diese werden wie die Positionsnummern in eckigen Klammern angegeben (eine Kombination mehrere Möglichkeiten ist ebenfalls erlaubt). Beachtet werden muss, dass für den Fall, dass keine Positionierungsnummer in der Zelle erwünscht ist, die «fehlende» Nummer mittels [ ] angegeben werden muss (siehe Beispiel 5).

**Beispiel 5a:** Übersicht aller vordefinierten Zellen

```
\begin{Puzzle}{5}{5}
|[8][f]X| [8][.]X| [8][*]X| [8][0]X| [8][o]X|.
|.
|[8][t]X| [8][T]X| [8][b]X| [8][B]X| [8][1]X|.
|.
|[8][L]X| [8][r]X| [8][R]X| [8][/]X| [8][,]X|.
\end{Puzzle}
```

**Beispiel 5b:** Angabe mehrerer Zelleigenschaften

```
\begin{Puzzle}{2}{2}
|[1][fT]X|.
|{}|[ ]|[T/r]X|.
\end{Puzzle}
```

Darüber hinaus ist es möglich, zusätzliche Zelltypen selber zu definieren. Hierfür stehen zwei Befehle zur Verfügung: Die Definition farbiger Zellen erfolgt mittels `\PuzzleDefineColorCell{1}{2}`; dieser Befehl stellt einen Spezialfall von `\PuzzleDefineCell{1}{2}` dar, mit dem Zellen allgemein verändert werden können. Das erste Argument enthält bei beiden Befehlen das Kürzel für die neu definierte Zelle, unter dem es in der `Puzzle`-Umgebung aufgerufen werden kann. Bei der Definition einer farbigen Zelle enthält das zweite Argument den Namen der Farbe; bei der allgemeinen Definition einer neuen Zelle enthält das zweite Argument die entsprechenden Befehle (siehe Beispiel 6).

<sup>1</sup>Das sind Rätsel, deren Lösungswörter nicht durch schwarze Kästchen getrennt werden sondern durch dicke Striche (manchmal werden diese ebenfalls zu den klassischen Kreuzwortsrätseln gezählt).

**Beispiel 6:** Neudefinition von Zellen

```
\PuzzleDefineCell{+}{%
  \PuzzleThickLine\put(#1,#2){%
    \framebox(1,1){}}
\PuzzleDefineColorCell{g}{lightgray}
\begin{Puzzle}{2}{2}
|[1][g]X|.
|{}      |[2][+]X|.
\end{Puzzle}
```

**Rahmen innerhalb eines Rätsels**

Oft sieht man in Zeitschriften, dass Kreuzwörterrätsel ein mittiges Bild oder eine Anzeige umschließen. Hierfür kann innerhalb der `Puzzle`-Umgebung der Befehl `\Frame{1}{2}{3}{4}{5}` verwendet werden, dessen fünf Argumente folgende Bedeutung haben: `{1}`, `{2}` bezeichnen die  $(x,y)$ -Koordinate der unteren linken Ecke des Rahmens. Wichtig hierfür ist, dass das untere linke Kästchen des Rätsels die Koordinaten  $(0,0)$  hat. Die Argumente `{3}`, `{4}` geben die Breite und Höhe des Rahmens in Zellen an. Das fünfte Argument enthält das Element (Text, Bild, ...), das innerhalb dieses Rahmens erscheinen soll. Zu beachten ist, dass die Zellen in der `Puzzle`-Umgebung, die dem Rahmen zugewiesen sind, als leer (`|{}|`) gekennzeichnet werden (siehe Beispiel 7).

**Beispiel 7:** Rahmen innerhalb des Rätsels

```
\begin{Puzzle}{10}{6}
\Frame{3}{2}{4}{2}{Having Fun \\ with \LaTeX{}}
|[1] A|[2] E|[3]G|   Y|[4] P|T |[5] E|[6] N|* |[7]A|.
|[8] P|   H|   I|* |   O|* |[9] S|   E|[10]I|   N|.
|   F|* |   P|{} |{} |{}|{} |   B|   N|* |.
|[11]E|[12]L| F|{} |{} |{}|{} |[13]E|   D| E|.
|[14]L|   O|   E|[15]H|[16]N|E|* |[17]L|   E|* |.
|* |[18]B| L|   A|   U|* |[19]C|* |[20]R| A|.
\end{Puzzle}
```

1	2	3		4		5	6		7
8						9		10	
11	12							13	
14				15	16			17	
	18						19		20

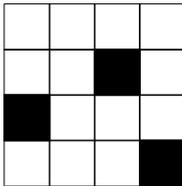
## Rätseltypen

Neben dem Standardkreuzwörtertsel, das bisher beschrieben wurde, bietet das `cpuzzle`-Paket noch die Möglichkeit, andere Rätseltypen zu setzen.

### Ausfüllrätsel

Für Ausfüllrätsel steht die `PuzzleWords`-Umgebung zur Verfügung, deren einziges Argument eine Ziffer enthält, die angibt wieviele Buchstaben die nun aufgeführten Lösungswörter besitzen (»Wörter der Länge `#[1]:«`). Innerhalb der Umgebung ist das Kommando `\Word{1}` definiert, in dessen Argument die einzelnen Wörter angegeben werden. (siehe Beispiel 8). Bei der Ausgabe der Lösung werden die Angaben in der `PuzzleWords`-Umgebung ignoriert.

#### Beispiel 8: Ausfüllrätsel



Worte der Länge 2: IN OB WO  
 Worte der Länge 3: EIN INN TON  
 Worte der Länge 4: OBEN WORT

```
\begin{Puzzle}{4}{4}
|W|O|R|T|.
|O|B|*|O|.
|*|E|I|N|.
|I|N|N|*|.
\end{Puzzle}

\begin{PuzzleWords}{2}
\Word{IN}
\Word{OB}
\Word{WO}
\end{PuzzleWords}
\begin{PuzzleWords}{3}
\Word{EIN}
\Word{INN}
\Word{TON}
\end{PuzzleWords}
\begin{PuzzleWords}{4}
\Word{OBEN} \Word{WORT}
\end{PuzzleWords}
```

### Sudoku

Mit dem Paket `cpuzzle` können seit der neuesten Version von diesem September auch Sudokus gesetzt werden. Ein Sudoku ist ein japanisches Rätsel, bei dem die Zahlen von 1 bis 9 verwendet werden – diese dürfen in jeder Zeile, jeder Spalte und jedem  $3 \times 3$ -Untergitter nur je einmal vorkommen. Um solche Rätsel zu setzen, wird die Umgebung `Sudoku` definiert, welche eine spezialisierte `Puzzle`-Umgebung darstellt, die auf  $9 \times 9$  Felder festgelegt ist und in der keine schwarzen Kästchen definiert sind. Der `*` wird in dieser Umgebung für Hinweiszahlen verwendet (siehe Beispiel 10).

## Beispiel 10: Sudoku

	2						9	
3		1	9		6	5		2
			8		4			
	9						5	
5			2		3			6
	7						2	
			4		7			
8		2	5		1	7		3
	5						8	

```
\begin{Sudoku}
| 7|*2| 4| 1| 3| 5| 6|*9| 8|.
|*3| 8|*1|*9| 7|*6|*5| 4|*2|.
| 9| 6| 5|*8| 2|*4| 1| 3| 7|.
| 2|*9| 6| 7| 1| 8| 3|*5| 4|.
|*5| 1| 8|*2| 4|*3| 9| 7|*6|.
| 4|*7| 3| 6| 5| 9| 8|*2| 1|.
| 6| 3| 9|*4| 8|*7| 2| 1| 5|.
|*8| 4|*2|*5| 9|*1|*7| 6|*3|.
| 1|*5| 7| 3| 6| 2| 4|*8| 9|.
\end{Sudoku}
```

## Zahlenrätsel

Um Zahlenrätsel zu setzen, stehen die beiden Befehle `\PuzzleLetters{1}` und `\PuzzleNumbers{1}` zur Verfügung. Der erste dient dazu, die verwendeten Buchstaben aufzulisten, der zweite gibt eine Reihe von Kästchen aus, in die der Rätselnde die gefundenen Nummern-Buchstaben-Kombinationen eintragen kann – im Argument dieses Befehls werden die Buchstaben in der Reihenfolge aufgelistet, wie sie den Nummern im Rätsel entsprechen. Bei der Ausgabe der Lösung werden das Rätsel sowie die zugehörigen Kombinationen gesetzt (siehe Beispiel 9).

## Beispiel 9: Zahlenrätsel

1	2	7	8
2	3		2
	6	4	5
4	5	5	

```
\begin{Puzzle}{4}{4}
|[1]W|[2]O|[7]R|[8]T|.
|[2]O|[3]B|*|[2]O|.
|*|[6]E|[4]I|[5]N|.
|[4]I|[5]N|[5]N|*|.
\end{Puzzle}
```

```
\PuzzleLetters{BEINORTW}
```

```
\PuzzleNumbers{WOBINERT}
```

Benutzte Buchstaben: BEINORTW

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

1	W	2	O	7	R	8	T	
2	O	3	B			2	O	
			6	E	4	I	5	N
4	I	5	N	5	N			

```

\puzzleSolution[true]
\begin{Puzzle}{4}{4}
|[1]W|[2]O|[7]R|[8]T|.
|[2]O|[3]B|*|[2]O|.
|*|[6]E|[4]I|[5]N|.
|[4]I|[5]N|[5]N|*|.
\end{Puzzle}

\PuzzleNumbers{WOBINERT}
    
```

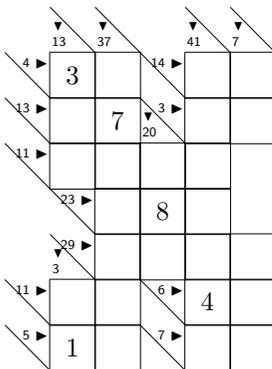
1	W	2	O	3	B	4	I	5	N	6	E	7	R	8	T
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Kakuro**

Ein anderer Rätseltyp aus Japan ist Kakuro. Hier ist das Ziel, Zahlen in einem Gitter so unterzubringen, dass sich je Zeile bzw. Spalte eine vorgegebene Summe ergibt. Erschwert wird die Aufgabe dadurch, dass sich die eingesetzten Zahlen einer Spalte bzw. Zeile in dieser nicht wiederholen dürfen.

Um solche Rätsel zu setzen, stellt das Paket `cwppuzzle` die Umgebung `Kakuro{1}{2}` zur Verfügung, deren zwei Elemente die Spalten- bzw. Zeilenzahl angeben. Innerhalb dieser Umgebung wird eine leere Zelle mit `| -` angegeben. Die Zellen, die die Summen enthalten, sind vom Typ `[,]`, d.h. sie enthalten eine diagonale Linie, aber keinen Rahmen. Die Summen selber werden in eckigen Klammern angegeben, getrennt durch einen Doppelpunkt, links von diesem steht die Summe, die für die jeweilige Spalte gilt, rechts die für die jeweilige Zeile. Erstere wird unterhalb der Diagonalen gesetzt, weitere oberhalb (siehe Beispiel 11)

**Beispiel 11: Kakuro**



```

\begin{Kakuro}{6}{9}
| - |<:13>|<:37>| - |<:41>|<:7>|.
|< 4:>| *3 | 1 |<14:>| 8 | 6 |.
|<13:>| 9 |*7 |<3:20>| 2 | 1 |.
|<11:>| 1 | 2 | 3 | 5 | - |.
| - |<23:>| 6 | *8 | 9 | - |.
| - |<29:3>| 8 | 9 | 7 | 5 |.
|<11:>| 2 | 9 |<6:>| *4 | 2 |.
|< 5:>| *1 | 4 |<7:>| 6 | 1 |.
| - | - | - | - | - | - |.
\end{Kakuro}
    
```

## Weitere Befehle

Für das »Feintuning« der gesetzten Rätsel stehen noch eine Reihe zusätzlicher Befehle zur Verfügung, die u. a. die verwendeten Schriftarten und -größen sowie Strichdicken betreffen. In der nachfolgenden Tabelle sind alle mit ihrer Bedeutung und ihrem Standardwert aufgelistet:

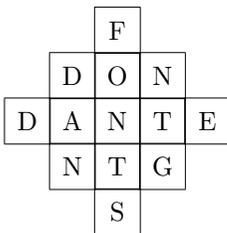
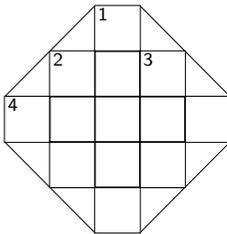
<i>Befehl</i>	<i>Bedeutung</i>	<i>Standard</i>
<code>\PuzzleUnitlength</code>	Länge einer Zelle	20pt
<code>\PuzzleClueFont</code>	Schrift der Aufgaben	<code>\footnotesize</code>
<code>\PuzzleFont</code>	Schrift innerhalb des Rätsels	<code>\rmfamily\normalsize</code>
<code>\PuzzleNumberFont</code>	Schrift der Positionierungsnummern	<code>\sffamily\scriptsize</code>
<code>\PuzzlePre</code>	Angabe von Befehlen, die vor der <code>Puzzle</code> -Umgebung ausgeführt werden sollen	neuer Absatz und zentrieren
<code>\PuzzlePost</code>	Angabe von Befehlen, die nach der <code>Puzzle</code> -Umgebung ausgeführt werden sollen	Absatzende und Ende der Zentrierung
<code>\PuzzleCluePre</code>	Angabe von Befehlen, die vor der <code>PuzzleClues</code> -Umgebung ausgeführt werden sollen	<code>minipage</code> -Umgebung mit 45% der Textbreite
<code>\PuzzleCluePast</code>	Angabe von Befehlen, die nach der <code>PuzzleClues</code> -Umgebung ausgeführt werden sollen	Ende der <code>minipage</code> -Umgebung
<code>\PuzzleHook</code>	Angabe weiterer Befehle für die <code>picture</code> -Umgebung, mit der das Kreuzgitter gezeichnet wird (Beispiel 12)	<code>\relax</code>
<code>\PuzzleBlackBox</code>	Angaben über die Art der schwarzen Felder (siehe Beispiel 13)	<code>\rule{% \PuzzleUnitLength}{% \PuzzleUnitlength}</code>
<code>\PuzzlePutNumber</code>	Positionierungsnummer – der Befehl hat 3 Argumente, das erste ist die $x$ -, das zweite die $y$ -Position der Zelle, das dritte gibt die Nummer mitsamt ihrer Positionierung an (z. B. <code>[t1]2</code> )	oben links in zugehöriger Zelle
<code>\PuzzleLineThickness</code>	Dicke der Zelllinien	.25pt

weiter auf der nächsten Seite...

<i>Befehl</i>	<i>Bedeutung</i>	<i>Standard</i>
<code>\PuzzleThickline</code>	Dicke einer fetten Linie (beispielsweise Trennstrich bei amerikanischen Kreuzwörterrätseln)	2pt
<code>\SudokuLinethickness</code>	Dicke der fetten Trennlinien der 3×3-Untergitter	2pt
<code>\KakuroNumberFont</code>	Schrift der Summen (Kakuro)	<code>\sffamily\tiny</code>
<code>\KakuroHintType</code>	Art der Zelle, die die Summen enthält (Kakuro)	[ , ]

Weitere Beispiele

**Beispiel 12:** Verwendung von `\PuzzleHook`



```

\newcommand\PuzzleHook{%
  \put(0,2){\line(1,-1){2}}%
  \put(0,3){\line(1,1){2}}%
  \put(5,2){\line(-1,-1){2}}%
  \put(5,3){\line(-1,1){2}}%
}
\begin{Puzzle}{5}{5}
|{} |{} |[1]F|.
|{} |[2]D| O|[3]N|.
|[4]D| A| N| T| E|.
|{} | N| T| G|.
|{} |{} | S|.
\end{Puzzle}

\renewcommand\PuzzleHook{\relax}
\PuzzleSolution
\begin{Puzzle}{5}{5}
|{} |{} |[1]F|.
|{} |[2]D| O|[3]N|.
|[4]D| A| N| T| E|.
|{} | N| T| G|.
|{} |{} | S|.
\end{Puzzle}

```

### Beispiel 13: Verwendung von `\PuzzleBlackBox`



```

\renewcommand{\PuzzleBlackBox}{%
  \rule{.75\PuzzleUnitlength}{.75\PuzzleUnitlength}}
\begin{Puzzle}{1}{1}
|*|.
\end{Puzzle}
\renewcommand{\PuzzleBlackBox}{%
  \framebox(.75,.75){\framebox(.5,.5){}}}
\begin{Puzzle}{1}{1}
|*|.
\end{Puzzle}

```

#### Hinweise

Für das Setzen von Sudokus finden sich auf CTAN noch zwei weitere Pakete. Zum einen `sudoku` von Paul Abraham, das eine ähnliche Syntax wie `cwpuzzle` aufweist, mit dem es jedoch nicht möglich ist, eine Lösungsausgabe automatisch zu erzeugen. Zum anderen finden sich drei verschiedene Pakete, die unter dem Namen `sudoko-bundle` zusammengefasst sind, von Peter Wilson. Diese sind wesentlich mächtiger als `cwpuzzle`, da sie nicht nur das Setzen von Sudokus erlauben, sondern diese auch lösen und in einem weiteren Schritt generieren.

## L<sup>A</sup>T<sub>E</sub>X im Unternehmen: Anwaltskanzlei

Alexander Willand

L<sup>A</sup>T<sub>E</sub>X wurde für die Wissenschaft konzipiert. Lässt sich L<sup>A</sup>T<sub>E</sub>X auch in Unternehmen in der sogenannten »Freien Wirtschaft« verwenden? Wie gut erfüllt L<sup>A</sup>T<sub>E</sub>X die Anforderungen an Software zur Erstellung von Texten in Unternehmen? Was funktioniert besser als bei Word & Co. und wo gibt es Probleme?

Der Verfasser dieses Artikels ist Anwalt auf dem Gebiet der erneuerbaren Energien und schildert hier seine Verwendung von L<sup>A</sup>T<sub>E</sub>X. Er hofft, dass in den nächsten Ausgaben dieser Zeitschrift andere »T<sub>E</sub>Xies« aus ihren Unternehmen berichten. Das könnte nicht nur interessant für Anwender sein, sondern auch für die Profis: Wie wird L<sup>A</sup>T<sub>E</sub>X in der Praxis genutzt?

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

Wie kommt man als Anwalt dazu, L<sup>A</sup>T<sub>E</sub>X zu lernen? Kurz gesagt, alles andere ist zu spannend: Kann man das Gutachten mit Graphiken, Tabellen und Berechnungen absturzfrei ausdrucken? Wird der Schriftsatz ans Gericht vielleicht um drei Versionen zurückgeworfen werden, weil das Textverarbeitungsprogramm klüger sein will als der Anwender? Vor fünf Jahren habe ich begonnen, längere Schriftstücke, bei denen ich wenig auf anderer Leute direkte Mitarbeit am Text angewiesen war, zu »texen«. Ich werde zunächst schildern, wie sich mein Arbeitsablauf entwickelt hat und welche Pakete ich verwende.

Im zweiten Teil dieses Artikels will ich auf die in den letzten fünf Jahren aufgetretenen Hindernisse eingehen, die sich bislang nicht beseitigen ließen.

### Die Tagesarbeit

Anwälte schreiben Briefe an Mandanten, an Behörden und Gerichte, an Geschäftspartner der Mandanten oder an böse Gegner. Wenn man nicht gerade eine »Nebelbombe« werfen will, kommt es stets darauf an, sich strukturiert und möglichst verständlich auszudrücken. Überraschenderweise ist Strukturierung bei den Briefklassen nicht eingeplant. Leslie Lamport hielt Briefe wohl nicht für den Hauptanwendungsbereich von L<sup>A</sup>T<sub>E</sub>X: In Briefen sind weder Gleitumgebungen noch Gliederungen vorgesehen. Beinahe hätte dies meine L<sup>A</sup>T<sub>E</sub>X-Karriere noch vor dem Start zur Bruchlandung gezwungen.

Als ich in einem Forum nach einer Gliederung für Briefe fragte, lautete die Antwort: »Schreib' doch die erste Seite als Brief und hänge den Rest als gesonderten Text, verfasst mit einer Artikel-Klasse, hinten an. Dann hast Du floats und sections.« Das wäre allerdings genau die Art von entnervender Komplikation, die man im Zeitdruck nicht braucht. Ich konnte weder mit derartigen Komplikationen leben, noch selbst T<sub>E</sub>X lernen, um Abhilfe zu schaffen.

Zum ersten Mal und viele weitere Male half Markus Kohm, der Autor von KOMA-Script, der kurzerhand seine Briefklasse um eine `sections.lco` ergänzte. Damit ist es seither möglich, Briefe durch Überschriften zu gliedern. Die erste Hürde war genommen, und Gleitumgebungen erwiesen sich in Briefen als entbehrlich.

Für mich als Anfänger war es natürlich schwierig, den Kanzlei-Briefkopf mit den Mitteln von `scr1ttr2` nachzubauen und das ganze dann unter Lyx bereitzustellen. Im Übrigen vereinfachte Lyx den Einstieg jedoch sehr. Auch heute schreibe ich die meisten Texte damit. Da die Oberfläche von Lyx intuitiv zu bedienen ist, kann ich sogar Briefe auf Band diktieren und von einer Mitarbeiterin tippen lassen.

Aber zurück zu den Anfängen. Ich habe damals zwei Dinge richtig gemacht: Bücher (eine Einführung in L<sup>A</sup>T<sub>E</sub>X und die KOMA-Script-Anleitung) gekauft und gelesen, sowie KOMA-Script verwendet. Wenn man vor den unvorhersehbaren Fehlern aus der Word-Welt flieht, dann wird die Toleranzgrenze für kryptische Fehlermeldungen ziemlich niedrig. KOMA-Script und Markus Kohm haben mich hier nie enttäuscht, die Software war stabil und der Maintainer reaktionsschnell.

Ich konnte meinen Mandanten – im übertragenen Sinn – nun gutes Vanilleeis in ungewöhnlichen Waffeln verkaufen, wollte aber auch Schokosauce über dem Eis und eine Kirsche obendrauf. Mein zweiter Schritt nach dem Einstieg – also der Fähigkeit, Briefe und Gutachten unfallfrei zu verfassen – bestand darin, eine Reihe von Helferlein zu suchen.

Eine große Vereinfachung brachte das Paket `microtype`. Damit gehörte in den Rand ragender Text der Vergangenheit an, soweit ich nicht vergaß, Bindestriche durch "=" zu ersetzen. Vorher musste ich jedesmal einen Brief absuchen, ob sich durch die letzten Änderungen irgendwo ein scheußlicher Umbruch ergeben hatte.

Vor zwei Jahren erschien eine Art Neuschöpfung der alten Times-Schrift unter dem Namen Linux Libertine. Ich war zu diesem Zeitpunkt gerade dabei, mit

ein paar Kollegen eine eigene Kanzlei zu gründen und so wurde die schöne Libertine die Kanzleischrift.

Aus der Vielzahl anderer unentbehrlicher und aufwändiger Pakete (`xcolor`, `hyperref`, `beamer`, ...) und Bücher (Begleiter, Praxisbuch, L<sup>A</sup>T<sub>E</sub>X-Hacks, Tabellen mit L<sup>A</sup>T<sub>E</sub>X) möchte ich ein besonders praktisches herausheben: `attachfile` von Scott Pakin ermöglicht es, aufgeräumte E-Mails zu versenden, weil alle Anhänge in das versendete PDF integriert werden. Die Kollegen von der Word-Fraktion versenden immer noch E-Mails mit Rumpelkeller-Anhängen.

Ich verfasse neben Briefen viele Verträge. Nach dem Scheitern meiner eigenen Versuche erbarmte sich Markus Kohm auch hier und programmierte das Paket `scrjura`, mit dem man Verträge entsprechend den hiesigen Gepflogenheiten abfassen kann. Es nummeriert Paragraphen und Abschnitte, wie es sich gehört. Es kann auf sämtliche 25 abgefahrenen Arten auf andere Paragraphen verweisen (auf § 4 Absatz 2 Satz 3 ebenso wie auf § 4 II 3). Man kann Tabellen und Formeln einsetzen und dabei die Absatzzählung nach Bedarf ein- und ausschalten. Schlichtweg großartig.

Voilà: Vanilleeis mit Schokosauce und Kirsche obendrauf. Und nicht Vanilleeis mit Schokosauce unten in der Waffel, mit im Eis versteckter Kirsche oder andersrum, je nach Laune von Word.

Im Lauf der Jahre kamen natürlich noch viele andere Helferlein dazu. Wie hängt man die aktuelle T<sub>E</sub>X-Datei an das daraus erstellte PDF, um sie zu versenden, nachdem sich eine Kollegin als »T<sub>E</sub>Xie« zu erkennen gegeben hatte? (`\expandafter\attachfile\expandafter{\jobname.tex}` – Danke nochmal an Heiko Oberdiek). Außerordentlich hilfreich war und ist das `mrunix`-Forum, in dem ich als »cookie170« mein Unwesen treibe. Auf `dctt` wurde ich hingegen oft für mein Pseudonym »Keks Dose« gerügt. Meine Entschuldigung an alle Helfer dort, aber meine (künftigen) Mandanten sollen, wenn sie mich googeln, nicht erst über zwei Seiten voller Meldungen zu »Latex« auf den Bildschirm rätseln müssen, wie dies bei der Verwendung meines Namens auf `dctt` zu befürchten wäre.

## Dauerbaustellen

Kommen wir zu den Dauerbaustellen. Unter Windows mit MikT<sub>E</sub>X und unter Linux auf dem Notebook mit T<sub>E</sub>X Live arbeiten? Kein Problem bei richtiger Kodierung, T<sub>E</sub>X ist das Betriebssystem egal. Schrifteninstallation und Briefkopfgestaltung: lösbare Aufgaben.

Auch die Einbindung in die Anwalts-Software erwies sich als überraschend einfach: Adressen lassen sich per copy & paste aus der Adressdatenbank übernehmen und die fertigen Dokumente in einem Verzeichnis »sonstige Dateien« (gedacht für z. B. von Mandanten gesendete Bilder, gescannte Dokumente) bei jeder elektronischen Akte speichern.

Aber dem gegnerischen Kollegen bei Vertragsverhandlungen eine Word-Version übersenden? Aufwändig. Nach einigen Experimenten erwies sich `tex4ht` als brauchbarste Lösung, aber es vermag die Paragraphenabsätze in Verträgen nicht zu erkennen. Häufig haben wir dann den Vertrag grob mit Word ausgehandelt und für die Feinarbeit, FreigabeprozEDUREN und Unterzeichnung habe ich dann daraus (über OpenOffice und dessen L<sup>A</sup>T<sub>E</sub>X-Export) eine L<sup>A</sup>T<sub>E</sub>X-Version gebaut. Inzwischen setzen sich PDFs mehr und mehr durch. Dank »PDF-Annotator« kann man in sie direkt hineinschreiben; dies erleichtert die Zusammenarbeit stark.

Auf Papier im DIN-A4-Format Text schön zu setzen, ist nicht so einfach. Der übliche einspaltige Satz führt zu überlangen Zeilen oder riesigen Rändern. Also verwende ich, wo immer möglich, zweisepaltigen Satz. In Verträgen oder Protokollen, die zweisepaltig abgefasst werden, treten an die Stelle des zweisepaltigen Fließtextes gegenüberliegende Spalten mit Originaltext und Übersetzung nebeneinander. Beides, Fließtext wie gegenüberliegende Spalten, haben ihre Tücken.

Am schönsten wäre es häufig, die Seite in eine breite Textspalte und eine schmale Marginalienspalte aufzuteilen. Es ist sehr bedauerlich, dass Marginalien, anders als Fußnoten, nicht über den Seitenumbruch hinweg auf der nächsten Seite fortgeführt werden können. Wenn jemand sich berufen fühlte, Marginalien den Seitenumbruch beizubringen, es fänden sich viele Abnehmer eines solchen Paketes.<sup>1</sup>

Bei zweisepaltigem Fließtext werden von Haus aus die beiden Kolumnen nicht ausgeglichen. Dem hilft zwar das Paket `multicol` ab. Leider setzt `multicol` die Fußnoten wieder nur einspaltig, anstatt die Fußnoten unter diejenige Spalte zu setzen, in der darauf verwiesen wird. Es wirkt seltsam, unter zweisepaltigem Text breite Fußnoten abzudrucken.

Als unerwartet schwierig erwiesen sich auch zweisepaltige Texte. Das Paket `parallel` bietet weder Überschriften, noch Aufzählungen oder Auflistungen, was sich nur notdürftig beheben lies. Leider muss man sich auch auf Schusterjungen und Hurenkinder einstellen, weil T<sub>E</sub>X mit `parallel` beim Seiten-

---

<sup>1</sup>Das Projekt »Tufte-LaTeX« (<http://code.google.com/p/tufte-latex/>) führt vor, wie gut das Layout durch derart asymmetrischen Satz werden kann, aber auch dort überstehen Marginalien den Seitenumbruch nicht.

umbruch sonderbare Entscheidungen trifft. Hiergegen ist noch kein Kraut gewachsen.

## Fazit

Der wahre Luxus für mich als Stressvielschreiber liegt darin, dass L<sup>A</sup>T<sub>E</sub>X keine Fehler bei Befehlen akzeptiert. Man kann mit Word schrecklichen Unfug anstellen, der einem Tage oder Wochen später einen wiederhol-, aber nicht beseitigbaren Absturz beschert. Bei L<sup>A</sup>T<sub>E</sub>X fällt einem jeder Unfug gleich auf, nicht erst fünf Minuten vor irgendeiner Deadline. Als nächstes werde ich versuchen, ein `Makefile` zu basteln, in dem unter Benutzung des Pakets `pdfpages` alle in einem Verzeichnis vorhandenen PDF-Dateien in ein umfassendes PDF gepackt werden; eine Vereinfachung also bei der Aktenführung.

# Von fremden Bühnen

---

## Neue Pakete auf CTAN

Jürgen Fenn

Der Beitrag stellt neue Pakete auf CTAN seit Mitte April 2009 bis zum Redaktionsschluss vor. Bloße Updates können auf der moderierten *CTAN-ann*-Mailingliste verfolgt werden, die auch über die Microblogging-Dienste *Identi.ca* und *Twitter* verfügbar ist.

`match_parens` von *Wybo Dekker* ist ein Perl-Skript, das dabei hilft, Fehler beim Einklammern von Ausdrücken zu finden.

CTAN:support/match\_parens

`latex-make` von *Vincent Danjean* enthält ein Makefile und weitere Tools, die beim Kompilieren von  $\text{\LaTeX}$ -Dokumenten behilflich sein können. GNU `make` 3.81 oder höher wird benötigt.

CTAN:support/latex-make

`rangen` von *D. P. Story* erzeugt Zufallswerte (ganze Zahlen, rationale Zahlen und Dezimalbrüche) auf der Grundlage des Pakets `lcg` von *Erich Janka*.

`eqexam` von *D. P. Story* dient zum Erstellen von Frage- und Testbögen, Quiz- und Hausaufgaben usw. (siehe auch S. 18)

`chklref` von *Jérôme Lelong* überprüft  $\text{\LaTeX}$ -Dokumente auf nicht (mehr) benutzte Labels.

CTAN:support/chklref

`widetable` von *Claudio Beccari* dehnt oder staucht alle Zellen einer Tabelle so weit, dass die Tabellenbreite insgesamt einen bestimmten Wert annimmt.

CTAN:macros/latex/contrib/widetable

`pst-abspos` von *Herbert Voß* ist ein neues `PSTricks`-Paket, mit dem man beliebige Objekte nach absoluten oder relativen Koordinaten auf einer Seite platzieren kann.

CTAN:graphics/pstricks/contrib/pst-abspos

`cachepic` von *Tomasz M. Trzeciak* dient dazu, Teile eines  $\text{\LaTeX}$ -Dokuments mithilfe eines Lua-Skripts automatisiert in EPS- und PDF-Dateien umzusetzen.

CTAN:macros/latex/contrib/cachepic

`mailmerge` von *Miguel V. S. Frasson* ist ein Paket zum Erstellen von Serienbriefen und anderen -drucksachen.

CTAN:macros/latex/contrib/mailmerge

`texdiff` von *Cengiz Gunay* erzeugt ein Dokument, in dem die Unterschiede zwischen zwei  $\text{\LaTeX}$ -Dokumenten hervorgehoben werden, ähnlich wie bei der

Änderungsverfolgung bei einer Textverarbeitung. Veränderungen werden im Seitenrand mit einem grauen Balken markiert.

CTAN:support/texdiff

**bracketkey** von *Christoph Heibl* erzeugt eingeklammerte Schlüssel zum Erstellen der Listen, in denen es auf den Genus der Items und auf andere Merkmale ankommt.

CTAN:macros/latex/contrib/bracketkey

**drv** von *Laurent Méhats* erzeugt Ableitungsbäume mit METAPOST. Kenntnisse in METAPOST werden nicht vorausgesetzt.

CTAN:graphics/metapost/contrib/macros/drv

**trimspaces** von *Will Robertson* erlaubt es, den Abstand bestimmter Zeichen, Argumente oder Makrodefinitionen zu angrenzenden Zeichen anzupassen.

CTAN:WillRobertson

**csv2latex** von *Alan Munn* ist eine Kombination aus einem Ruby- und einem Applescript, mit der man Tabellen aus Microsoft Excel, OpenOffice.org Calc und iWork Numbers in einen L<sup>A</sup>T<sub>E</sub>X-Quelltext kopieren kann. Unterstützt werden einfache Tabellen sowie die Pakete **booktabs** und **longtable**. Funktioniert mit T<sub>E</sub>XShop und jedem anderen applescriptfähigen Editor ab Mac OS X 10.4.

CTAN:support/csv2latex

**memdesign** von *Peter Wilson* ist eine erweiterte Fassung des ersten Teils der Anleitung zur **memoir**-Klasse, die vom Rest des Manuals getrennt wurde, weil sie auch für Anwender interessant sein dürfte, die mit anderen Klassen arbeiten.

CTAN:info/memdesign

**dox** von *Didier Verna* ist eine Erweiterung zu **doc**, mit der man nicht nur Makros und Umgebungen, sondern auch Optionen, Zähler usw. beschreiben kann.

CTAN:macros/latex/contrib/dox

**ltxcmds** von *Heiko Oberdiek* stellt einige Makros aus dem L<sup>A</sup>T<sub>E</sub>X-Kernel in einem anderen Namensraum auch für andere Formate bereit, beispielsweise für Plain-T<sub>E</sub>X.

CTAN:macros/latex/contrib/oberdiek

**tablenotes** von *Matthias Borck-Elsner* dient dazu, Fußnoten wie Endnoten in Tabellen zu setzen.

CTAN:macros/latex/contrib/tablenotes

**listings-ext** von *J. Hoffmann* dient zur automatisierten Übernahme von Quelltext in ein L<sup>A</sup>T<sub>E</sub>X-Dokument.

CTAN:macros/latex/contrib/listings-ext

**threeparttablex** von *Lars Madsen* erweitert **threeparttable**, um Fußnoten auch in langen Tabellen setzen zu können.

CTAN:macros/latex/contrib/threeparttablex

**font-change** von *Amit Raj Dhawan* enthält 19 Plain-T<sub>E</sub>X-Makros, um die Text- und die Mathematik-Schriftart und die Schriftgröße mit einem einzigen Befehl ändern zu können.

CTAN:macros/plain/contrib/font-change

**jflap2tikz** von *Andrew Mertz* wandelt eine JFlap-Datei in eine L<sup>A</sup>T<sub>E</sub>X-Datei. Der dargestellte Automat wird mit Hilfe von TikZ gezeichnet.

CTAN:graphics/jflap2tikz

**garrigues** von *Denis Roegel* enthält METAPOST-Makros, um *Garrigues'* Oster-Nomogramm zu erzeugen (vgl. TUGboat volume 30, number 1, 2009, p. 88–104).

CTAN:graphics/metapost/contrib/macros/garrigues

**suanpan** von *Denis Roegel* enthält METAPOST-Makros zum Zeichnen japanischer und chinesischer Abakusse (vgl. TUGboat volume 30, number 1, 2009, p. 74–79).

CTAN:graphics/metapost/contrib/macros/suanpan

**spverbatim** von *Scott Pakin* erlaubt Zeilenumbrüche innerhalb von `\verb` sowie in der `verbatim`-Umgebung.

CTAN:macros/latex/contrib/spverbatim

**titlepic** von *Thomas ten Cate* dient zum Einbinden einer Abbildung auf der Titelseite eines Dokuments.

CTAN:macros/latex/contrib/titlepic

**boolexpr** von *Florent Chervet* dient zum Verarbeiten von booleschen Ausdrücken.

CTAN:macros/latex/contrib/boolexpr

**germkorr** von *Timo Baumann* korrigiert das Kerning, so dass einige Buchstaben, z. B. das T, dadurch näher an die deutschen Anführungszeichen heranrücken.

CTAN:macros/latex/contrib/germkorr

**presentations** von *Herbert Voß* enthält die Beispiele aus dem Buch »Präsentationen mit L<sup>A</sup>T<sub>E</sub>X« aus der Edition *date*.

CTAN:info/examples/Presentations\_de

**fig4latex** von *Joe Fields* erzeugt ein Makefile, mit dem man Abbildungen in großen Projekten leichter verwalten kann.

CTAN:graphics/fig4latex

**mathgifg** von *Boris Veytsman* enthält die L<sup>A</sup>T<sub>E</sub>X-Unterstützung für die Fonts *Microsoft Georgia* und *ITC Franklin Gothic* sowohl für den Text- als auch für den Mathematik-Modus. Die Schriften werden mit Microsoft Windows und Microsoft Office 2008 ausgeliefert.

CTAN:fonts/mathgifg

**ltxnew** von *Florent Chervet* stellt die Befehle `\new`, `\renew` und `\provide` bereit, die vor der Definition eines neuen Befehls testen, ob der Befehlsname schon vergeben ist.

CTAN:macros/latex/contrib/ltxnew

**changelayout** von *Ahmed Musa* ändert das Layout einzelner Seiten. Es ist eine Erweiterung von `changepage`.

CTAN:macros/latex/contrib/changelayout

**keycommand** von *Florent Chervet* beruht auf dem Paket `kvsetkeys` von *Heiko Oberdiek* und erlaubt das Definieren von Befehlen mit optionalen *keys*.

CTAN:macros/latex/contrib/keycommand

`lshort-persian` von *Mehdi Omidali* ist die persische Übersetzung von `lshort-german`.

Es wurde mit  $X_{\text{L}}^{\text{A}}\text{TeX}$  und `bidif` erstellt.

CTAN:info/lshort/persian

`biblatex-authoryear-icomp-tt` von *Thomas Titz* ist ein neuer `biblatex`-Stil für das Autor-Jahr-Schema mit kompakten Mehrfachzitaten und *ibidem*-Ersetzung.

CTAN:macros/latex/exptl/biblatex-contrib

`etextools` von *Florent Chervet* stellt nützliche  $\epsilon\text{-TeX}$ -Makros für  $\text{L}^{\text{A}}\text{TeX}$ -Anwender und Paketautoren zur Verfügung.

CTAN:macros/latex/contrib/etextools

`properties` von *Michael Niedermair* lädt Eigenschaften im Key-Value-Format aus einer externen Datei.

CTAN:macros/latex/contrib/properties

`context-top-ten` von *Yue Wang* ist eine elementare Einführung in `ConTeXt`, in der die wichtigsten Befehle vorgestellt werden.

CTAN:info/context-top-ten

`chinese-font-metrics` von *Yue Wang* sind Font-Metriken zur Verwendung mit chinesischen Fonts. Näheres erfahren des Chinesischen Kundige unter <http://bbs.ctex.org/viewthread.php?tid=50078>.

CTAN:fonts/chinese-font-metrics

`asymptote-by-example` von *Yue Wang* ist ein chinesisches Asymptote-Tutorium.

CTAN:info/asymptote-by-example

`asymptote-manual-zh` von *Yue Wang* ist die chinesische Übersetzung des Asymptote-Handbuchs.

CTAN:info/asymptote-manual-zh

`context-notes-zh-cn` von *Yue Wang* ist ein chinesisches `ConTeXt`-Tutorium.

CTAN:info/context-notes-zh-cn

`asy-faq-zh-cn` von *Yue Wang* ist die englische Übersetzung der Asymptote-FAQ von <http://asymptote.sourceforge.net/FAQ>.

CTAN:info/asy-faq/zh-cn

# Bücher und Rezensionen

---

## Edition dante – Neuauflage

Herbert Voß:

**L<sup>A</sup>T<sub>E</sub>X Referenz**; 2. Auflage 2010,  
DANTE e. V. und Lehmanns Media  
240 Seiten (ISBN 978-3-86541-366-6;  
ca. 14,95 € (Ladenpreis) bzw. 12,- € für Mit-  
glieder von DANTE e.V., jeweils versandkos-  
tenfrei)



## Bestellung

Bitte schicken Sie eine E-Mail an [office@dante.de](mailto:office@dante.de) mit Angabe von *Name*, *Anschrift*, *Mitgliedsnummer* und *Anzahl der Exemplare*, und überweisen Sie den Betrag auf das Konto von DANTE e.V. oder bezahlen Sie per PayPal. Die Kontonummer finden Sie am Ende dieses Heftes und Informationen zu PayPal auf <http://www.dante.de/index/Intern/Zahlung.html>.

Bitte beachten Sie für Bestellungen bei DANTE e.V. folgende Informationen zum Widerrufsrecht: Verbraucher können bei Bestellungen per E-Mail, Internet, Brief oder Telefon den Kaufvertrag innerhalb einer Frist von 14 Tagen ab Erhalt der Ware per Brief, Fax oder E-Mail oder durch Rücksendung der Ware widerrufen (siehe Kontaktadresse). Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs oder der Ware. Bei einem Bestellwert bis 40,- € hat der Besteller die Rücksendekosten zu tragen. Bei Verschlechterung der Ware, die über die übliche Prüfung der Ware hinausgeht, hat der Besteller gegebenenfalls Wertersatz zu leisten.

## »Abschlussarbeiten und Präsentationen mit L<sup>A</sup>T<sub>E</sub>X« von Paul Resetarics

Christine Römer

Die Publikation »Abschlussarbeiten und Präsentationen mit L<sup>A</sup>T<sub>E</sub>X« stellt die Installation von MacT<sub>E</sub>X und MikT<sub>E</sub>X vor und geht auf das Setzen von Abschlussarbeiten und Präsentationen ein.

Als ich kürzlich im Medienkaufhaus Dussmann in Berlin – nach der Eigenwerbung das »renommierteste Medienkaufhaus in Deutschland« – mal die Angebote zu L<sup>A</sup>T<sub>E</sub>X in Augenschein nahm, fiel mir zum einen negativ auf, dass nicht ein Exemplar der von DANTE e.V. mit herausgegebenen Publikationen zu T<sub>E</sub>X und L<sup>A</sup>T<sub>E</sub>X im Angebot war. Zum anderen war ich erfreut, einen Titel »Abschlussarbeiten und Präsentationen mit L<sup>A</sup>T<sub>E</sub>X« vorzufinden, der den barocken Untertitel trägt »Manual zur Verwendung des Textsatzprogramms L<sup>A</sup>T<sub>E</sub>X beim Verfassen wissenschaftlicher Abschlussarbeiten und Präsentationen unter den Betriebssystemen Microsoft Windows und Apple Macintosh OS X«. Der Autor Paul Resetarics, der auch den Satz besorgt hat, wird als Lehrer für Gesundheits- und Krankenpflege vorgestellt. Diese Informationen und ein kurzes Durchblättern ließen die Hoffnung keimen, dass dies eine Publikation sein könnte, die ich bei meinen Studierenden aus einem universitären, geisteswissenschaftlichen Bereich in den L<sup>A</sup>T<sub>E</sub>X-Einführungen einsetzen könnte. Da der Installation breiter Raum eingeräumt wird, verstärkte sich dieser Eindruck. Gerade die meist als schwierig empfundene Installation ist häufig der Grund, dass die Studierenden nach dem Kurs nicht bei L<sup>A</sup>T<sub>E</sub>X bleiben. Die Freude über die Entdeckung dieser Einführung währte allerdings nicht lange und machte dem Ärger über die von dieser Publikation verpassten Chancen Platz. Bei den weitläufigen Erläuterungen zum T<sub>E</sub>XnicCenter (S. 31-44) wird nur die Deklaration des Ausgabeprofiles T<sub>E</sub>X → PDF besprochen und als ausreichend behauptet, was nicht einsichtig ist. Gerade für Erweiterungen wie beispielsweise `eps2eps` wäre eine Hilfestellung nützlich. Auch wäre das Eingehen auf die Handhabung des Rechtschreibkorrekturprogramms recht sinnvoll gewesen, da es für die angesprochenen Schüler und Studierenden wichtig ist. Besonders ist zu bedauern, dass nicht auf die von DANTE e.V. vertriebenen



Zusammenstellungen (T<sub>E</sub>X Collection) verwiesen wird, wo ja mit proT<sub>E</sub>Xt eine einfache Installationsvariante unter Microsoft Windows bereitgestellt wird, die alle notwendigen Komponenten (T<sub>E</sub>X-System, Editor, Konverter und Viewer) enthält.

Eine Begründung, die ich den Studierenden für den Einsatz von L<sup>A</sup>T<sub>E</sub>X immer gebe, ist die Möglichkeit, damit einen typografisch schönen Text abliefern zu können. Diese muss man mit dem Manual in der Hand weglassen, da der Satzspiegel so wenig gelungen ist<sup>1</sup> wie die eingefügten, teilweise verschwommen aussehenden Abbildungen. Auch wirbt das Buch nicht für L<sup>A</sup>T<sub>E</sub>X, wenn es eine zweite Auflage weder zu Kolumnentiteln noch zu einem Index gebracht hat und Abkürzungen ohne `thinspace` gesetzt wurden.

Leider lässt sich über die Behandlung der beiden Schwerpunkte Abschlussarbeiten und Präsentationen wenig Lobenswertes schreiben, beide kranken daran, dass der aktuelle T<sub>E</sub>Xnische Stand wenig berücksichtigt wird. Wieso für die Zielgruppe deutschsprachige Anfänger gar nicht auf das komplexe Paket KOMA-Script<sup>2</sup> verwiesen wird, ist nicht nachvollziehbar. Dies trifft bei den Präsentationen auf die Klasse `powerdot` zu, zu der es neuerdings auch eine Übersetzung der Dokumentation ins Deutsche gibt. Stattdessen werden neben `beamer` »Overhead-Folien mit `slide`« behandelt.

Für den Themenschwerpunkt Präsentationen sei auf das sehr gelungene Buch »Präsentationen mit L<sup>A</sup>T<sub>E</sub>X« (2009) von Herbert Voß in der Arbeitsheftreihe von DANTE e.V. verwiesen, das (fast) keine Wünsche offen lässt. Die gesamte DANTE-Reihe »Arbeitshefte für T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X u. a.« wird von P. Resetarics nicht erwähnt. Es erfolgt auch kein Hinweis darauf, dass es sinnvoll ist, auf die Paketdokumentationen zurückzugreifen.

Paul Resetarics

Abschlussarbeiten und Präsentationen mit L<sup>A</sup>T<sub>E</sub>X

Facultas Verlags- und Buchhandels AG: Wien 2009 (2. Auflage)

151 Seiten, inkl. DVD, 17 × 24 cm

ISBN 978-3-7089-0418-4, 21,30 €

---

<sup>1</sup>Dies trifft auch – wenngleich nicht so extrem – auf »Der L<sup>A</sup>T<sub>E</sub>X-Tutor« von I. Hedtke, D. Gipper, R. Müller (Shaker Media: Aachen 2009) zu, wo besonders der sehr breite äußere und der viel zu knappe untere Rand auffallen, der immer etwas größer als der obere sein sollte.

<sup>2</sup>Markus Kohm / Jens-Uwe Morawski: KOMA-Script – Die Anleitung. DANTE e.V., Lehmanns Media, 3. Auflage. 2008

# Spielplan

---

## Termine

3. 3. – 5. 3. **DANTE 2010**  
und 42. Mitgliederversammlung von DANTE e.V.  
Universität Dortmund  
<http://www.dante.de/events/dante2010.html>

13. 3. – 14. 3. **Chemnitzer Linxstage 2010**  
Technische Universität Chemnitz  
09107 Chemnitz  
<http://www.chemnitzer.linux-tage.de>

28. 6. – 30. 6. **TUG 2010**  
San Francisco (Kalifornien), USA  
<http://tug.org/tug2010/>



(Fotos: Martin Sievers)

## 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/index/Stammtische.html>.*

### Aachen

Torsten Bronger  
bronger@physik.rwth-aachen.de  
Gaststätte Knossos, Templergraben 28  
Zweiter Donnerstag im Monat, 19.00 Uhr

### Berlin

Rolf Niepraschk  
Tel.: 0 30/3 48 13 16  
Rolf.Niepraschk@gmx.de  
Gaststätte Bärenschenke  
Friedrichstraße 124  
10117 Berlin Mitte  
Zweiter Donnerstag im Monat, 19.00 Uhr

### Bremen

Winfried Neugebauer  
Tel.: 04 21-8 28 65 14  
tex@wphn.de  
Wechselnder Ort  
Erster Donnerstag im Monat, 18.30 Uhr

### Darmstadt

Karlheinz Geyer  
geyerk.fv.tu@nds.tu-darmstadt.de  
Hostaria Toskana  
Nieder-Ramstädter-Straße 13  
64283 Darmstadt  
<http://www.da-tex.org>  
Erster Freitag im Monat, ab 19.30 Uhr

### Dortmund

Martin Schröder  
martin@oneiros.de  
Tel.: 0231-1206574  
Wechselnder Ort  
Zweiter Mittwoch im Monat, 18.30 Uhr

### Dresden

Carsten Vogel  
lego@wh10.tu-dresden.de  
Studentenwohnheim, Borsbergstraße 34,  
Dresden, Ortsteil Striesen  
ca. alle 8 Wochen, Mittwoch, 19.00 Uhr

### Düsseldorf

Georg Verweyen  
Georg.Verweyen@web.de  
Bistro/Café Zicke

Böckerstr. 5 a (Ecke Bergerallee)  
40213 Düsseldorf  
Zweiter Mittwoch in ungeraden Monaten,  
20.00 Uhr

### Erlangen

Walter Schmidt, Peter Seitz  
w.a.schmidt@gmx.net  
Gaststätte »Deutsches Haus«  
Luitpoldstraße 25  
Dritter Dienstag im Monat, 19.00 Uhr

### Freiburg

Heiko Oberdiek  
Tel.: 07 61/4 34 05  
oberdiek@uni-freiburg.de  
<http://www.inacker.de/TeX/>  
Wechselnder Ort  
Dritter Donnerstag im Monat, 19.30 Uhr

### Hamburg

Lothar Fröhling  
lothar@thefroehlings.de  
Zum Schwarzenberg  
Schwarzenbergstr. 80 – 21073 Hamburg  
Letzter Dienstag im Monat, 19.30 Uhr

### Hannover

Mark Heisterkamp  
heisterkamp@rrzn.uni-hannover.de  
Seminarraum RRZN  
Schloßwender Straße 5  
Zweiter Donnerstag im Monat, 18.30 Uhr

### Heidelberg

Martin Wilhelm Leidig  
Tel.: 06203/402203  
moss@moss.in-berlin.de  
Anmeldeseite zur Mailingliste:  
<http://mailman.moss.in-berlin.de/mailman/listinfo/stammtisch-hd-moss.in-berlin.de>  
»Goldener Hirsch«  
Hauptstr. 18, 68526 Ladenburg  
Letzter Freitag im Monat, ab 19.30 Uhr

### Karlsruhe

Klaus Braune  
Tel.: 07 21/6 08 40 31  
braune@rz.uni-karlsruhe.de

Universität Karlsruhe, Rechenzentrum  
Zirkel 2, 3. OG, Raum 316  
Erster Donnerstag im Monat, 19.30 Uhr

### Köln

Helmut Siegert  
Institut für Kristallographie  
Zülpicher Straße 49b  
Letzter Dienstag im Monat, 19.30 Uhr

### München

Uwe Siart  
uwe.siart@tum.de  
[http://www.siart.de/typografie/  
stammtisch.xhtml](http://www.siart.de/typografie/stammtisch.xhtml)  
Gaststätte »Marktwirt«  
Heiliggeiststr. 2  
Erste Woche des Monats an wechselnden  
Tagen, 19.00 Uhr

### Stuttgart

Bernd Raichle  
bernd.raichle@gmx.de  
Bar e Ristorante »Valle«  
Geschwister-Scholl-Str. 3  
Zweiter Dienstag im Monat, 19.30 Uhr

### Trier

Martin Sievers  
stammtisch-trier@texberatung.de  
Fetzenkneipe (Haus Fetzenreich)  
Sichelstraße 36 (beim Sieh-Um-Dich)  
Dritter Montag des Monats, 20.15 Uhr

### Ulm

Adelheid Grob  
latex@heidi.in-ulm.de  
<http://latex.in-ulm.de>  
Ratskeller Ulm (Rathaus Ulm)  
Dritter Montag im Monat, 19.30 Uhr

### Wuppertal

Andreas Schrell  
Tel.: 02193/53 10 93  
as@schrell.de  
Restaurant Croatia »Haus Johannisberg«  
Südstraße 10  
an der Schwimmoper Wuppertal-Elberfeld  
Zweiter Donnerstag im Monat, 19.30 Uhr

### Würzburg

Bastian Hepp  
LaTeX@sning.de  
nach Vereinbarung



(Fotos: Martin Sievers)

# 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., 10.00–12.00 Uhr)  
Fax: 0 62 21/16 79 06  
E-Mail: [dante@dante.de](mailto:dante@dante.de)

Konto: VR Bank Rhein-Neckar eG  
BLZ 670 900 00  
Kontonummer 2 310 007  
IBAN DE67 6709 0000 0002 3100 07  
SWIFT-BIC GENODE61MA2

## Präsidium

Präsident:	Klaus Höppner	<a href="mailto:president@dante.de">president@dante.de</a>
Vizepräsident:	Volker RW Schaa	<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:	Manfred Lotz	<a href="mailto:secretary@dante.de">secretary@dante.de</a>
Beisitzer:	Günter Partosch Bernd Raichle Herbert Voß	

## Server

CTAN: <http://mirror.ctan.org>  
DANTE: <http://www.dante.de/>

## Autoren/Organisatoren

- Jürgen Fenn** [72] **Idris Samawi Hamid** [27]  
 Friedensallee 174/20  
 63263 Neu-Isenburg  
 juergen.fenn@gmx.de  
 Colorado State University  
 ishamid@colostate.edu
- Karlheinz Geyer** [15] **Volker RW Schaa** [4]  
 Kirchstraße 16  
 64665 Alsbach  
 geyerk.fv.tu@nds.tu-darmstadt.de  
 siehe Seite 82
- Jürgen Gilg** [18] **Martin Schröder** [7]  
 Austraße 59  
 70376 Stuttgart  
 gylt@oneiros.de  
 Firma QuinScape GmbH  
 Wittekindstraße 30  
 44139 Dortmund  
 martin@oneiros.de
- Adelheid Grob** [15, 54] **Herbert Voß** [3, 76]  
 Sedanstraße 67  
 89077 Ulm  
 latex@heidi.in-ulm.de  
 Wasgenstraße 21  
 14129 Berlin  
 herbert@dante.de
- Klaus Höppner** [4, 7] **Dominik Wagenführ** [12]  
 siehe Seite 82  
 [9] dwagenfuehr@freiesmagazin.de
- Manfred Lotz** [40, 50] **Alexander Willand** [67]  
 Schriftführer von DANTE e.V.  
 manfred@dante.de  
 Wichernstraße 2  
 76185 Karlsruhe  
 alexander.willand@t-online.de
- Herbert Möller** [77] **Uwe Ziegenhagen** [36]  
 Elsternweg 10  
 48329 Havixbeck  
 mollerh@math.uni-muenster.de  
 Lokomotivstr. 9  
 50733 Köln  
 uwe@ziegenhagen.info
- Christine Römer**  
 Institut für germanistische  
 Sprachwissenschaft  
 FSU Jena  
 Christine.Roemer@uni-jena.de

# Die T<sub>E</sub>Xnische Komödie

---

21. Jahrgang Heft 4/2009 Dezember 2009

## Impressum

### Editorial

#### Hinter der Bühne

- 4 Grußwort
- 7 DANTE 2010 – Einladung zur Mitgliederversammlung und »Call for Papers«
- 9 Beschlüsse der 41. Mitgliederversammlung von DANTE e.V. am 12. September 2009 in Esslingen

#### T<sub>E</sub>X-Theatertage

- 12 DANTE e.V. in Esslingen
- 16 Kieler Linux- und Open-Source-Tage am 2. und 3. Oktober 2009

#### Bretter, die die Welt bedeuten

- 18 Aufgabenblätter, Tests, Klausuren, ...
- 27 OpenType Engineering in T<sub>E</sub>XSummary Report for the Oriental T<sub>E</sub>X Project
- 36 Dynamisches Ausblenden von Text und Erstellung von Lückentexten
- 40 L<sup>A</sup>T<sub>E</sub>X-Figuren mit Hilfe von OpenOffice.org 3 Draw
- 50 Von pageref zu hyperpage
- 54 (Kreuzwort-)rätsel setzen mit L<sup>A</sup>T<sub>E</sub>X
- 67 L<sup>A</sup>T<sub>E</sub>X im Unternehmen: Anwaltskanzlei

#### Von fremden Bühnen

- 72 Neue Pakete auf CTAN

#### Bücher und Rezensionen

- 76 Edition dante – Neuauflage
- 77 »Abschlussarbeiten und Präsentationen mit L<sup>A</sup>T<sub>E</sub>X«  
von Paul Resetarics

#### Spielplan

- 79 Termine
- 80 Stammtische

#### Adressen

- 83 Autoren/Organisatoren