Beispieldokument zum Projekt *Multimediale Lehre* an der Technischen Universität Graz

Andreas Hirczy*
Winfried Kernbichler†
David Camhy

22. Februar 2008

Inhaltsverzeichnis

*Inst. für Theoretische Physik, Tel.: +43/3 16/8 73-81 90; ahi@itp.tugraz.at
†Inst. für Theoretische Physik, Tel.: +43/3 16/8 73-81 82; winfried.kernbichler@itp.tugraz.at

1 Grundlagen

1.1 Bedürfnisse

Dem am Institut für Theoretische Physik der TU Graz gewählten Ansatz für die Erstellung und Präsentation von Lehrinhalten liegt die Erfüllung folgender Bedürfnisse von Lehrenden und Studierenden zu Grunde:

- Einbindung möglichst vieler Lehrender und Studierender in die Erstellung der Inhalte und Beiträge
- Erstellung konsistenter Inhalte
  - am eigenen Webserver
  - in Präsentationsunterlagen
  - in Skripten und Druckvorlagen
  - am E-Learning Server
  - in Beispielen für Übungen und Praktika
- Vernünftige Koexistenz von klassischen Lehrveranstaltungen (Vorlesungen, Übungen, Praktika, …) mit einer Präsenz im Internet
- Verwendung möglichst weniger Masterdokumente
- Leichtigkeit der Erstellung
- Verwendung von Programmier- und Skriptsprachen, die im Bereich der Theoretischen Physik verbreitet sind
- Verwendung bestehender Infrastruktur
- Möglichkeiten zur Modifikation und zum weiteren Experimentieren und Programmieren für die Studierenden
- Leichte Abgabe von Übungsbeispielen in allen Übungen mit Computereinsatz

1.2 Lösungen

Dadurch haben wir uns im Wesentlichen zu folgender Vorgangsweise entschieden:

- Automatisierte Generierung von Einstiegs-Webseiten aus einfachen Dokumenten
- Verwendung von LaTeX, `latex2html` und pdf-LaTeX zur Erstellung der Texte
- Erstellung von Skripts zur Automatisierung unter Linux und UNIX mit Perl und Python

- Erstellung von Beispielen in Zusammenarbeit zwischen Vortragenden und Studierenden ohne externe Auftragnehmer
- Verwendung von Programmiersprachen, die im Studienplan "Technische Physik" angeboten werden
  - MATHEMATICA - wenn symbolische Berechnungen gewünscht oder notwendig sind
  - MATLAB - zur Erstellung von numerischen Simulationen inklusive graphischer Benutzeroberflächen
  - C++/Python - für numerische Simulationen mit QT oder TK/Inter als Benutzeroberfläche
  - C++/FORTRAN 95 als numerisches Backend, wenn MATLAB zu langsam ist
- Verwendbarkeit alle Programme unter Linux und MS Windows
- Einschränkungen
  - MATHEMATICA benötigt zumindest das Programm MATHREADER
  - MATLAB: zum gegenwärtigen Zeitpunkt benötigt man eine MATLAB-Installation mit Lizenz, da der MATLAB Compiler nicht den benötigten Umfang an Befehlen bewältigt.
- Weitere Elemente für Webseiten und Dokumente
  - Exportierte Animationen aus MATLAB und MATHEMATICA
  - Beschränkter Einsatz von JAVA Programmen

2 Einführung

Wir gehen davon aus, daß der langfristige Erfolge von „Multimediales Lernen im WEB“ nur gesichert sein kann, wenn der Zusatzaufwand für die Autoren der Lernbehelfe möglichst gering ist. Das bezieht sich auch auf die Bedienung der zu erstellenden Werkzeuge, vor allem aber auf die zugrundeliegenden Inhalte selbst.

Deswegen sollten unsere Methoden folgenden Kriterien genügen:

- Alle Präsentationsformen sollen aus einem Master-Dokument generiert werden. Damit soll die Konsistenz von Lernmaterial am Kursserver, eigenen HTML-Seiten, Skripten und Folien sichergestellt werden.
- Alle Vorgänge sollen soweit als möglich scriptgesteuert ablaufen können, um manuelle Eingriffe zu vermeiden und den nötigen Zeitaufwand zu minimieren. Nebenbei dient auch diese Vorgangsweise der Erhaltung der Konsistenz, da einzelne Schritte der Verarbeitung nicht vergessen werden können.

Wir haben uns für die Kombination von LaTeX, `latex2html` und pdfLaTeX entschieden, weil es vor allem im Bereich der theoretischen Physik und Mathematik seit vielen Jahren die beliebteste Anwendung zum Erstellen von Dokumenten ist und daher sowohl entsprechende Kenntnisse als auch Textmaterial verfügbar ist.

2.1 Verfügbarkeit und Dokumentation

- LaTeX-Kurzanleitung, Bücher von Kopka (alte Auflagen behandeln LaTeX 2.09, nicht LaTeX2e), LaTeX-Companion ... liegen im Institut auf und können entlehnt werden.
- Zugang zu Dokumentation für LaTeX im Internet erhalten sie über den Webserver des Instituts für theoretische Physik auf der Seite `http://www.itp.tu-graz.ac.at/links-software.ht`
- Auf allen Linux-Computern des Instituts und im Computerraum Physik sind aktuelle und soweit sinnvoll vollständige TeX-Implementierungen installiert.
- Dieser Text ist über den WWW-Server des Instituts (`http://www.itp.tu-graz.ac.at/`) auch im LaTeX Quellformat zugänglich, um als Vorlage für die Erstellung eigener Texte im Rahmen des Projekts *Multimediale Lehre* zu dienen.
- Die in diesem Dokument angesprochenen und verwendeten Erweiterungen sind auf den Linux-PCs des Instituts und im Computerraum Physik bereits installiert und im Verzeichnis extras/ dieses Beispieldokuments enthalten.

2.2 Allgemeine Hinweise zur Benutzung von LaTeX

Ich beginne mit einer kleinen Anleitung für den sauberen und effizienten Umgang mit LaTeX, damit sollte es kaum Probleme in der Zusammenarbeit mit dem Electronic Learning Server http://wbt.tu-graz.ac.at/ geben.

- Wir verwenden LaTeX2e; die Hauptdokumente beginnen daher mit `\documentclass[]{}`.
- Wann immer möglich, sollen fertige Pakete aus dem LaTeX-Standardpaket verwendet werden - die Chance, daß latex2html und damit zurecht kommt ist wesentlich höher
- Einbinden von Hyperlinks mit dem Makropaket HYPERREF; Die Dokumentation dazu ist in /usr/share/texmf/doc/latex/hyperref/manual.pdf
- Bibliographie immer im BibTeX-Format schreiben (siehe Kopka)
- Achtung bei DATEINAMEN: Groß/Kleinschreibung ist bei Windows nicht wichtig, bei Linux ergeben sich dann Probleme
- beim Schreiben mit LINUX und EMACS verwendet man am besten die vorinstallierte Konfigurationsdatei `~/.emacs`, dann erhalten sie automatische die Unterstützung für AUCTEX, REFTEX, BIBTEX.
- *form follows function* — nicht zu früh mit Spezialitäten des Layouts beginnen; LaTeX bedeutet logische Markup - Feintuning komm erst am Ende (wenn dann noch Zeit bleibt)
- spezielle Layoutwünsche des Vortragenden bitte hinterfragen – manche kleinen Veränderungen machen größere Mühe als in plainTeX oder MS Word, haben aber oft keine Bedeutung („ich will die Kapitelüberschriften 2 mm näher am Text haben“)
- Referenzen auf die Online-Hilfen der Mathematik-Programme werden durch selbstdefinierte LaTeX-Befehle erleichtert: TODO

  ```
  \matlabref{...}
  \mapleref{...}
  \mathematicaref{...}
  ```

  Die genaue Syntax dieser Befehle ist derzeit nicht spezifiziert.

3 Präsentationsformen

In diesem Beispiel wird gezeigt, wie alle angeführten Präsentationsformen erzeugt werden können. Bitte sehen sie in den Quellen und hier vor allem im Makefile nach, wie das exakt gelöst wird.

3.1 Druck- und Kopiervorlagen

Zur Herstellung von Druckvorlagen verwenden Sie am besten die übliche Kombination von `latex` und `dvips`; Sie umgehen damit alle Probleme, die beim Drucken über den *Acrobat Reader* auftreten können.

3.2 Skriptum am Webserver - PDF

Verwenden sie zum Erstellen der PDF-Dateien das Programm `pdflatex`; es verhält sich im Allgemeinen wie `latex`, allerdings sind die verwendbaren Bildformate andere. Im Abschnitt 6 wird auf diesen Unterschied näher eingegangen.

Um die Vorteile von PDFzu nützen wird folgende Sequenz in die Präambel übernommen:

```
\usepackage[colorlinks,breaklinks,pagebackref,pdfstartview=FitH]{hyperref}
\hypersetup{
  pdftitle={Beispieldokument zum Projekt Multimediale Lehre},
  pdfauthor={Andreas Hirczy, Winfried Kernbichler},
  pdfsubject={},
  pdfkeywords={},
  baseurl={http://itp.tugraz.at/},
  linkcolor={blue}
}
```

Diese Einstellungen sorgen dafür, daß die Metainformationen der Zieldatei ausgefüllt werden; sie können diese Daten im *Acrobat Reader* unter dem Menüpunkt [File][Document Info][General] sehen. Bitte entnehmen sie Details der Dokumentation zu pdflatex und hyperref.

3.3 Skriptum am Webserver - HTML

Aus den LaTeX-Quellen kann mit dem Programm latex2html der Text auch in eine HTML-Repräsentation übergeführt werden. Problematisch ist dabei neben grundsätzliche Problemen mit der Darstellung mathematischer Ausdrücke in HTML vor allem auch eine Schwäche des Konverters, komplexe LaTeX-Ausdrücke wie Makrodefinitionen … korrekt zu transformieren.

Die aktuelle verwendete Version von latex2html (2K.1beta 1.59) hat in userer Installation außerdem Probleme mit der Erzeugung mancher Formeln, die als Bilder eingebunden werden.

Für die Verwendung als Online–Skriptum scheint daher die Variante PDF besser geeignet.

3.4 Generierung von geteilten PDF-Dateien

Zum Erzeugen geteilter PDF-Dateien („Lerneinheiten") wird das Perl-Programm *pdflatex-parts* verwendet. Vom Hauptdokument werden alle Teile mittels des LaTeX-Befehls `\include` referenziert; das Programm erzeugt dann die entsprechenden PDF-Fragmente im Verzeichnis `pdf-parts/`.

3.5 Erzeugen von Präsentationsvorlagen für Videobeamer

Wenn der Kopf der LaTeX-Sourcen anstelle von

```
\documentclass[11pt,a4paper]{article}
```

oder ähnlichem mit der Sequenz

```
\ifx\MMLFOLIE\empty
  \typeout{## Erzeugen von FOLIEN / PRAESENTATIONSUNTERLAGEN}
  \documentclass[12pt,a5paper,landscape]{article}
\else
  \typeout{## Erzeugen der DRUCKVORLAGE}
  \documentclass[11pt,a4paper]{article}
\fi
```

beginnt, wird bei einem „normalem" Aufruf (`$ pdflatex beispiel.tex`) das Skriptum in einer Form erzeugt, die für den Druck und zum Lesen am Schirm geeignet ist. Wenn aber LaTeXin der Form

```
$ pdflatex '\def\MMLFOLIE{} \input beispiel.tex'
```

aufgerufen wird, wird der erste Teil der if-Verzweigung gewählt und für kleineres, querformatiges Papier gesetzt. Die *fullscreen*-Darstellung im *acrobat reader* sorgt dann für eine gut lesbare Darstellung in der Videoprojektion.

3.6 Generierung von Index-Seiten

Zum Erzeugen der Indexseiten liest das Perl-Programm `mml-make-index.pl` den Inhalt der Datei README und produziert eine Vorlage, die mit dem Programmpaket Website

Maker Language an das Erscheinungsbild der Webseiten am ITP angepaßt wird. Durch die Befehlsfolge

```
$ mml-make-index.pl | wml > index.html
```

kann damit die Titelseite zu den Teilprojekte erzeugt werden.

Das Programm erkennt selbständig, ob Dokumentation im PDF oder HTML-Format vorliegt und legt Links darauf an.

Die Datei README besteht im wesentlichen aus Tags und erklärendem Text; die Tags werden durch Einfügen in „curly braces“ gekennzeichnet. Der Rest ein Zeile mit Tag wird ignoriert. Ein Vorlage wird mit dem Archiv zu diesem Dokument mitgeliefert.

Derzeit werden folgende *Tags* erkannt:

Projekt Name des Projekts, insbesondere der Dateinahme der Dokumentation ohne die Erweiterung.

Titel Titel des Projekts, wie er in der Übersicht und Projektbeschreibung erscheinen soll.

Autoren Namen der Autoren; einzelne Autoren durch Kommas getrennt, jeweils mit Vor- und Nachnamen

Werkzeug Implementierungssprachen

Wichtig wichtige oder dringende Anmerkungen, werden auf des Titelseite rot hinterlegt

BeschreibungKurz eine kurze Beschreibung für die Projektübersicht

BeschreibungLang eine ausführlichere Beschreibung für die Titelseite; wenn das Projekt eine ergänzende Programmierarbeit zu einem anderen Projekt oder Skriptum ist, kann dies die gesamte Dokumentation sein.

4 LaTeX Erweiterungen

4.1 Hyperref Unterstützung für LaTeX2html

Das Makropaket `hyperref` aus pdfLaTeX wird teilweise für LaTeX2html unterstützt. Bitte konsultieren sie mich bei Problemen.

4.2 Hilfsfunktionen für LaTeX

> Diese Erweiterung sind derzeit nicht implementiert; im Lauf der nächsten Wochen ist mit allgemeiner Verfügbarkeit zu rechnen.

Um einfacher Hyperlinks auf Funktionen der verwendeten Programmiersprachen und Werkzeuge zu verwenden, werden einige spezielle Makros definiert:

- matlabref
- mathematicaref
- mapleref

Geplant ist außerdem ein Befehl zum Setzen von Textblöcken oder mathematischen Ausdrücken mit Titeln; die Syntax wird vermutlich in der Form `\framedblock{Titel}{Inhalt}` sein.

Zur Verwendung dieser Befehle ist die Einbindung des Paketes `mml` notwendig:

```
\usepackage{mml}
```

5 Beispiele für mathematischen Formelsatz

Wir beginnen mit einem einfachen Beispiel

$$e = mc^2$$

und versuchen danach eine Fomel, die zur besonderen Hervorhebung eingerahmt ist:

$$\boxed{e = mc^2}$$

5.1 Dirac-Notation — braket

Zur Verwendung der Dirac-Notation in der Quantenmechanik wurde die Stiloption braket installiert. Das Makropaket wird mit `\usepackage{braket}` nach `\documentclass...` im Hauptdokument eingebunden.

Beispiele für `\bra{} \ket{} \braket{} \set{}`:

$$\langle\Phi| \quad |\Phi\rangle \quad \langle\Phi|O|H|\Phi\rangle \quad \{\, \Phi \mid \hat{O}|H|\Phi \,\} \tag{1}$$

Beispiele für `\Bra{} \Ket{} \Braket{} \Set{}`:

$$\langle\Phi| \quad |\Phi\rangle \quad \left\langle \frac{\Phi}{1} \middle| O|H|\Phi \right\rangle \quad \left\{\, \Phi \;\middle|\; \hat{O}|H|\Phi \,\right\} \tag{2}$$

6 Verwenden von Bildern

Wir unterscheiden anhand der günstigsten Form der Speicherung drei verschiedene Bildtypen:

EPS — objektorientierte Grafiken bestehen aus Linien und Flächen, die durch die Koordinaten von Kontrollpunkten definiert werden. Diese Abbildungen können deshalb ohne Qualitätsverlust beliebig vergrößert werden. Das hauptsächliche Anwendungsgebiet liegt in selbsterzeugten Diagrammen.

Postscript bietet als vollständige Programmiersprache ein breite Palette von Möglichkeiten Unfug zu treiben; interessanterweise sind gerade die großen und teuren Programmpakete wie Mathematika oder Matlab hier besonders anfällig.

JPG — photographische Rasterbilder werden im allgemeinen durch technische Erfassung von Vorlagen durch Scanner oder digitale Fotoapparate erzeugt. Diese Abbildungen können im JPG-Format effizient gespeichert werden.

PNG — grafische Rasterbilder sind z.B. gescannte Diagramme oder Screenshots. Bei diesen Bildern erzeugt JPG-Kompression deutliche Artefakte, weil harte Kanten gegenüber fließenden Farbtonverläufen überwiegen

Durch den Aufruf von `LaTeX-pictures` werden diese Bildformate so ineinander umgewandelt, daß sowohl LaTeX als auch pdfLaTeX damit umgehen können.

6.1 Tricks

- Dateien sollen möglichst ohne den weißen Rand, den viele Programme gern einfügen, verarbeitet werden.

  Bei den Rasterformaten PNG und JPG verwenden sie zum *croppen* am besten die Graphikprogramme `xv` oder `gimp`. Bei PostScript-Dateien kann der Rand mit dem Befehl `epsffit`, `ps2epsi` oder `eps2eps` entfernt werden. Diese Programm beseitigen auch noch kleine Problem im Dateiaufbau; weil die erzeugenden Programme leider alle ihre eigenen Macken haben, müssen Sie hier durch Versuch und Irrtum die richtige Vorbehandlung selbst herausfinden.

- Verwenden Sie `\includegraphics` statt `\epsfig` zum Laden Ihrer Abbildungen.

- Geben Sie Dateinamen immer ohne *extension* an, damit LaTeX und Verwandte selbst den richtigen Dateityp wählen können.

6.2 Implementierung

Das Programm `LaTeX-pictures` verwendet folgende Verfahren zum Wandeln der Formate:

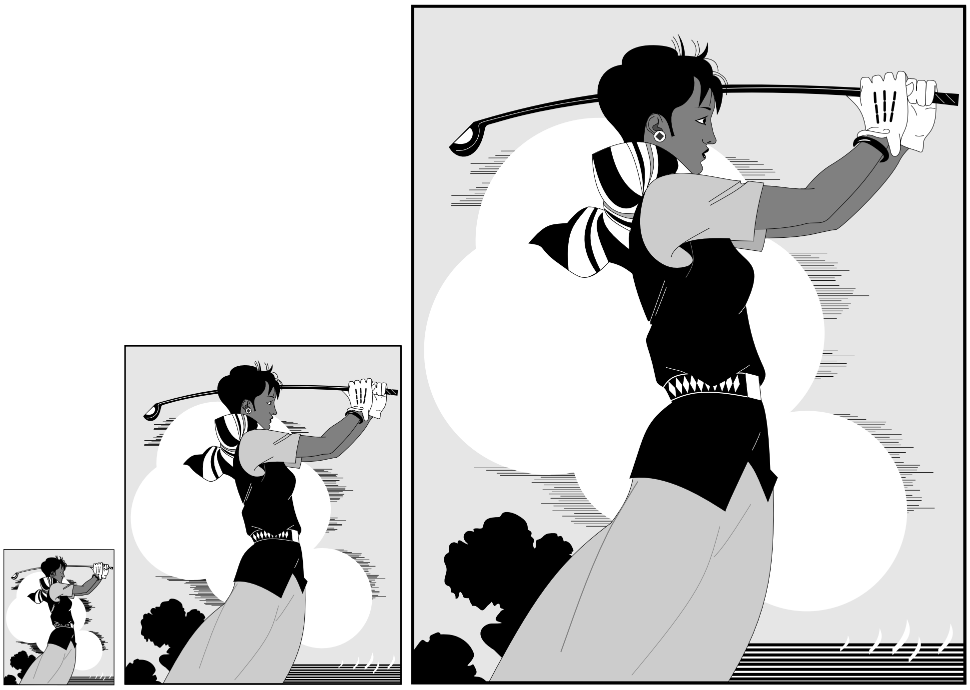

Abbildung 1: Beispiel für die Einbindung einer objektorientierten Grafik (das Bild golfer.ps kommt als Demonstrationszeichnung mit GhostScript)

Abbildung 2: Beispiel für die Einbindung einer digitalen Photographie

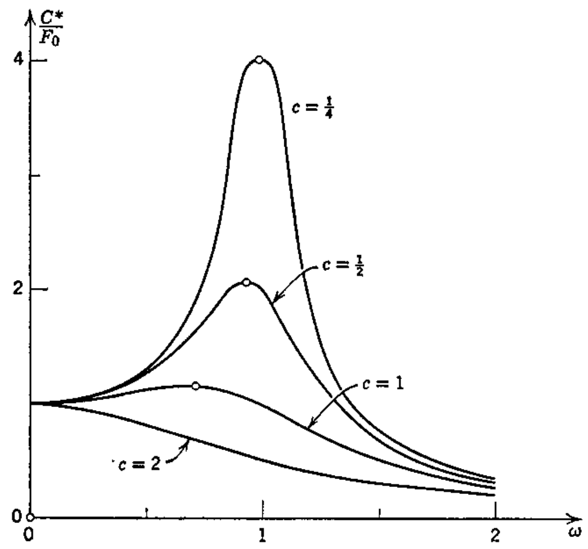

Abbildung 3: Beispiel für die Einbindung einer gescannten Grafik (aus [Kre72, S. 92])

JPG → EPS Konvertierung über `jpeg2ps (1)` `http://www.pdflib.com/jpeg2ps/`; dabei bleibt die Kompression des JPG-Formates erhalten.

PNG → EPS Konvertierung über `sam2p (1)` (`http://www.inf.bme.hu/~pts/sam2p/`)

EPS → PDF Konvertierung über `epstopdf (1)` (Teil von teTEX) `ftp://ftp.dante.de/tex-archive/syst`

Die Umwandlungen werden nur durchgeführt, wenn die zu erzeugende Datei nicht existiert oder älter als die Quelle ist. Die Konvertierung von EPS → PDF wird nur vorgenommen, wenn weder eine Datei im JPG oder PNG Format vorhanden ist, um Mehrdeutigkeiten nach der Konvertierung von PNG/JPG → EPS → PDF zu vermeiden.

6.3 Pathologische Beispiele

In diesem Abschnitt werden Bilder gesammelt, an denen Probleme mit früheren Versionen der MML-Tools demonstriert werden können.

6.4 MetaPost

Der wesentliche Vorteil von MetaPost liegt in der guten Integration in TEX und LATEX und der hohen Qualität der Ausgabe. MetaPost generiert Grafiken im Format PostScript mit so einfacher Struktur, daß sie auch von PDF-Interpretern verstanden werden können. Es eignet sich deshalb vor allem für die Darstellung von Diagrammen.

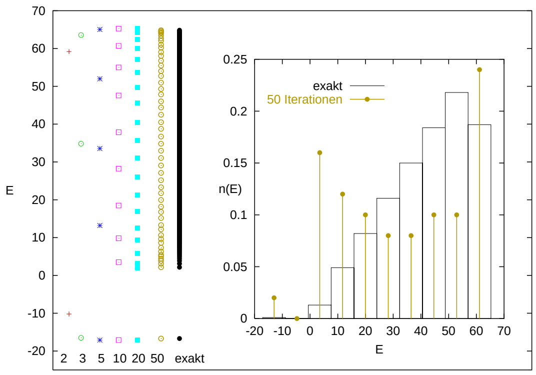

Abbildung 4: Maria Daghofer: mymaten.eps – Die „bounding box“ der aus EPS generierten PDF Dateien war nicht korrekt. Dies konnte durch Verwendung von „epstopdf“ anstelle von „ps2pdf“ als Konverter behoben werden.

Problematisch ist manchmal die Erzeugung der Bilder, weil die Kommandosprache von MetaPost zwar sehr umfangreich und vielseitig ist, aber die Erstellung von Grafiken aus Textdateien für manche Menschen etwas ungewohnt ist. Manche Zeichenprogramme (zB. xfig, tgif ???, metadraw) können allerdings nach MetaPost exportieren.

So macht man ein Bild mit MetaPost:

- Erstellen der MetaPost Vorlage `mpost diagramm.mp` mit xfig/vi/emacs/.... Sie können für Beschriftungen die übliche mathematische Notation von LaTeX verwenden.
- Umwandeln in PostScript mit `mpost diagramm.mp`
- Umbenennen der entstandenen Datei `diagramm.0` in `diagramm.mps`
- Einbinden in den LaTeX-Source mit `\includegraphics{diagramm}`
- Verabeiten der LaTeX-Source mit `latex datei; dvips datei` oder `pdflatex`.

Die von `mpost` erzeugten Bilder können prinzipiell mit jedem Interpreter für PostScript betrachtet werden, allerding ist die Beschriftung noch nicht korrekt aufgebaut; für einen Voransicht verwenden Sie am besten die Kommandofolge

```
tex mproof *.mps && dvips mproof && gv mproof.
```

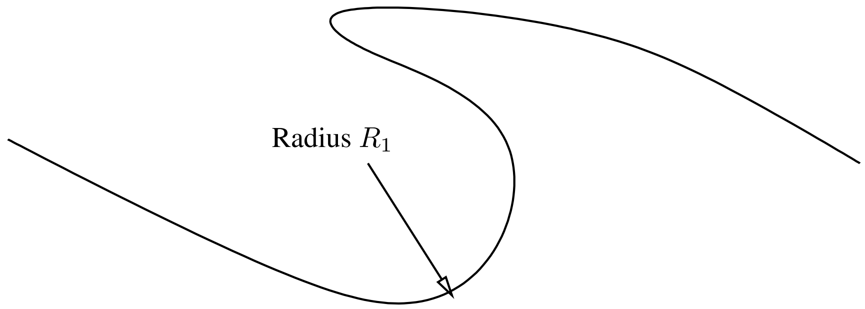

Abbildung 5: Beispiel für ein Diagramm im Format MetaPost

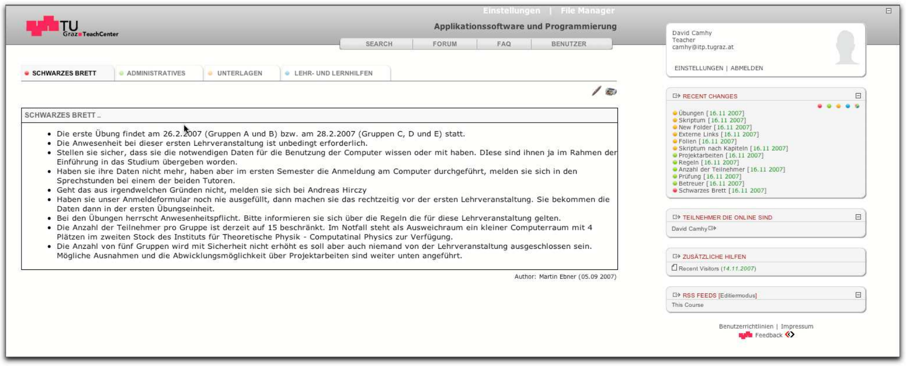

Abbildung 6: Schwarzes Brett

7 Interaktion mit dem TeachCenter

Sie können zum TU Graz TeachCenter über ihre Visitenkarte im TUGOnline (Rubrik: Forschung & Lehre) gelangen Eine praktische Abkürzung, die direkt zu Ihren Kursen führt, ist `http://tugtc.tugraz.at/wbtmaster/tutRoom.htm`.

Das TeachCenter ist aus Sicht des Autors in 3 Hauptabschnitte geteilt:

Schwarzes Brett Das schwarze Brett (6) ist für Ankündigungen und dergleichen vorgesehen; Änderungen führen sie mit Ihrem Web-browser direkt aus.

Administratives Dieser Abschnitt (7) wird im wesentlichen aus den Daten des TUG-Online gespeist; Interaktion sind möglich, aber nicht unbedingt notwendig.

Unterlagen Hier können Vorlesungsunterlagen, Übungszettel, Beispielprogramme, Testdaten ... abgelegt werden; das Programm `wbt_latex` soll Ihnen helfen, diese Bereiche einfach mit Inhalten zu füllen.

7.1 Vorbereitung der Unterlagen

Übersetzen Sie ihre LaTeX-Quellen mit `pdflatex` und `latex2html`; danach können Sie mit `wbt_latex` die notwendigen XML-Dateien[1] für den Import in das TeachCenter erzeugen – gleichzeitig werden Ihre Daten auf unseren Webserver installiert. Details zum Umgang mit `wbt_latex` können Sie durch den Aufruf mit dem Parameter `wbt_latex --man` erfahren.

Sie können zusätzliche Dateien oder Links ins Internet in einer Datei `MANIFEST` eintragen; auf diese wird ebenfalls in einer XML-Datei verwiesen und sie werden gegebenfalls auf den Webserver geladen. Das erste Wort in dieser Datei kennzeichnet dabei die Datei oder

[1]XML ist eine Konvention zum Speichern strukturierter Daten

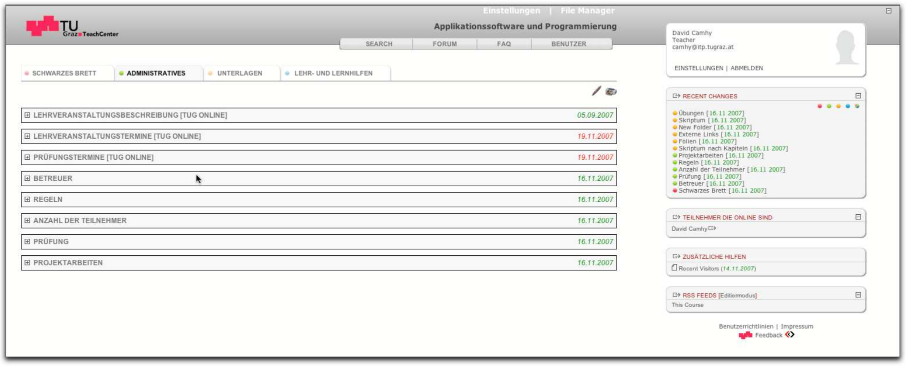

Abbildung 7: Administratives

einen Link; die restlichen Worte werden als Beschreibung verwendet. Eine Beispieldatei sehen Sie in `http://itp.tugraz.at/LV/ahi/beispiel/MANIFEST`.

7.2 Import in das TeachCenter

Aktivieren Sie im Kurs den Knopf „Filemanager“ oben rechts und gehen dann weiter zu „Import Library“. In das erscheinende Editierfeld kopieren Sie per „cut and paste“[2] einfach den gesamten Inhalt (`cat *.xml`) der erzeugten XML-Dateien.

Danach sind die Skripten und Zusatzdateien in Ihrem Kurs unter dem Karteireiter „Unterlagen“ verfügbar.

7.3 Textänderungen im TeachCenter

Zur Bearbeitung der Bereiche „Schwarzes Brett“ und „Administratives“ steht ihnen ein *Editierknopf* (stilisierte Füllfeder, siehe Abb. 8) zur Verfügung. Mit **Neu** und **Edit** können Sie neue Bereiche erzeugen und bestehende Bereiche bearbeiten.

Die Bearbeitung erfolgt im HTML Format. Um die Reihenfolge der Bereiche zu ändern, steht ihnen unten ein **Auswahlfeld** zur Verfügung. Mit **Upload File** können Sie eine zusätzliche Datei hochladen, zu der dann ein Verweis eingeblendet wird, den sie in ihrem HTML-Code verwenden können. Mit einem Klick auf **HTML** öffnet sich ein einfacher Editor, der ihnen eine direkte Vorschau des von ihnen bearbeiteten Bereiches zeigt.

Bei der Übernahme der Daten aus dem TUG-Online werden bestimmte Layouts verwendet. Wir hoffen, dass in Zukunft der Zugriff auf diese Vorlagen einfacher gestaltet wird. Wenn Sie bis dahin dieses Erscheinungsbild nachahmen möchten, können Sie folgende Konstruktion verwenden:

[2] Für die Zukunft ist eine benutzerfreundlichere und besser automatisierbare Art des Imports geplant.

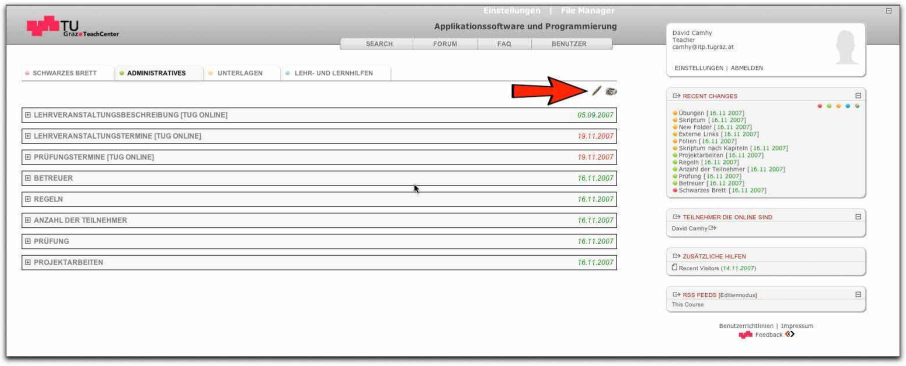

Abbildung 8: Editierknopf

Überschrift einfach:

```
<h1>Überschrift</h1>
```

Überschrift wie im TUG-Online - Import:

```
<div summary="ll_newspaper_importer 603"
style="font-family:arial,san-serif;font-size:15px;width:75%;
border-top-width:0;border-left-width:0;border-right-width:0;
border-bottom-width:3px;border-color:#a5b6c6;
border-style:solid;padding:1px;
margin-bottom:5px;margin-top:5px;"><b><i>
Überschrift
</i></b></div>
```

Text einfach:

```
<p>Text ...</p>
```

Text wie im TUG-Online - Import:

```
<div summary="ll_newspaper_importer 615" style="margin-left:20px;
margin-top:5px;padding-left:20px;">
Fließtext
</div>
```

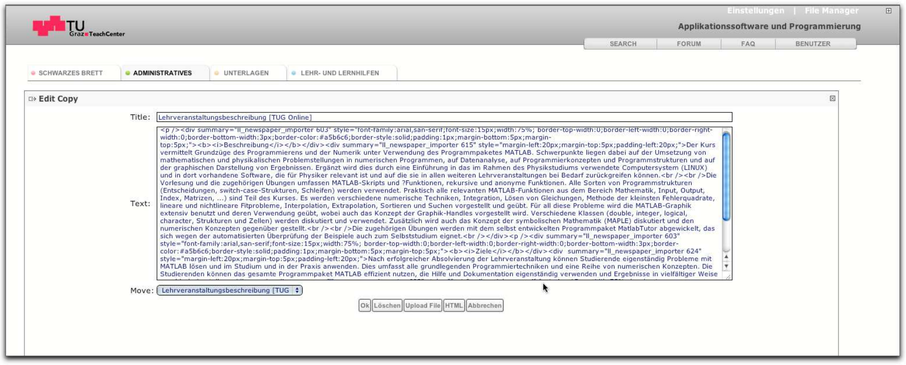

Abbildung 9: Bearbeitung eines Bereiches

Abbildung 10: Einfacher HTML Editor

Literatur

[Kre72] Kreyszig, Erwin: *Advanced Engineering Mathematics*. 3. New York : John Wiley and Sons, 1972 14

[RS98] Rossani, Alberto ; Spiga, G.: Kinetic Theorie with Inelastic Interactions. In: *Transport Theory and Statistical Physics* (1998), Nr. 27, S. 237–287