

L^AT_EX im (Hoch-) Schulalltag

Prof. Dr. Georg Lachenmay
Fachhochschule Rosenheim

25. Februar 2001

Im (Hoch-)Schulalltag fallen einige (größere) Dokumente an, wie Skripten, Diplomarbeiten, Berichte, (Kurz)-Referate usw., die gut mit den Standardklassen erstellt werden können. Ähnliches gilt für den „Papierkram“, wie Briefe, Faxe, Aushänge usw.

Dann gibt es aber noch einige Dinge, die man weder mit den weitverbreiteten WYSIWIG-Systemen, noch mit L^AT_EX-Standardklassen vernünftig erstellen kann. Das sind z.B. Stundenpläne oder Prüfungsaufgaben mit Musterlösungen. Dafür gibt es zwar spezielle L^AT_EX-Pakete die aber nicht optimal an den Bedarf an einer deutschen (Hoch-)Schule angepaßt sind.

Sehr wichtig ist auch die Erstellung von Folien für die OHP-Projektion oder die Präsentation mit dem Beamer bzw. die Herstellung von Web-Dokumenten für Internet oder Intranet.

Der Vortrag soll in aller Kürze die Klassen vorstellen, ihre Vor- und Nachteile sowie Ansatzpunkte für eine Weiterentwicklung aufzeigen.

1 Warum überhaupt L^AT_EX an einer (Hoch-) Schule?

In diesem Kreis zu erörtern, warum man L^AT_EX an (Hoch-) Schulen sinnvoll einsetzen kann – das hieße Eulen nach Athen tragen. Dennoch sollen hier einmal die Vorteile und Nachteile kurz gegenübergestellt werden:

1.1 Vorteile von L^AT_EX

Die bekannten Vorteile von L^AT_EX sind:

- überragende typografische Qualität des gedruckten Ergebnisses, unabhängig vom Ausgabegerät,
- Plattform-Unabhängigkeit,
- hohe Stabilität, weil ausgereift,
- geringe Ansprüche an Systemressourcen (ohne Grafik),
- sehr weite Kompatibilität zu früheren Versionen,

- weitgehend Zwang zum logischen Mark-up,
- Fähigkeit zur Bearbeitung auch sehr großer Dokumente,
- Wiederverwendbarkeit des Geschriebenen für OHP-Folien etc,
- Programmiermöglichkeiten zur „intelligenten“ Dokumentengestaltung.

1.2 Nachteile von \LaTeX

Dem stehen die ebenfalls bekannten Nachteile gegenüber:

- keine Klick-Bedienung (=„unmodern“),
- Einarbeitung aufwendig (wird zumindest so empfunden),
- Einschränkung der freien Gestaltung der Dokumente,
- keine Lobby.

1.3 Folgerung für (Hoch-) Schulen

Wenn man den Bildungsauftrag eines Gymnasiums oder einer Fachhochschule in Sachen *Methodenwissen Computer* ernst nimmt, dann kann es nicht Ziel sein, den Absolventen die Bedienung eines mehr oder weniger zufällig verfügbaren Office-Paketes zu vermitteln. Vielmehr sollte es Ziel sein, das *logische Strukturieren* von Dokumenten zu lehren und das dann praktisch umzusetzen.

Viele Schulen und auch Hochschulen haben Probleme damit, die immer schneller werdenden *Update-Zyklen* mitzumachen, weil diese enorme Kosten für Software-Updates und meist auch Hardware-Upgrades verschlingen. \LaTeX (eventuell unter Linux) könnte da ein Ausweg sein, weil da auch ein Pentium 100 oder gar ein 486er noch gut Dienste tut – so können die spärlichen Investitionsmittel sinnvoller eingesetzt werden. . .

Aber auch die *typografische Qualität* des Ergebnisses sollte Beachtung finden. Oft verbringen Schüler und Studenten sehr, sehr viel Zeit, weil Sie mit dem Ergebnis von gängigen Office-Produkten nicht zufrieden sind, und daher zu „basteln“ anfangen. Dieser Zeitverlust wiegt die (vermeintlich) aufwendige Einarbeitung in \LaTeX oft mehrfach auf.

Doch nun zum Alltag. Welche Dokumente fallen an und wie kann man Sie mit \LaTeX sinnvoll erstellen?

2 „Große“ Textdokumente

Im (Hoch)-Schulalltag gibt es zahlreiche Dokumente, die als „groß“ bezeichnet werden können. Das bedeutet, daß sie

- rund 50–200 Seiten haben können,
- eine logische Struktur aufweisen,
- zusätzliche Verzeichnisse, wie Inhaltsverzeichnis, Literaturverzeichnis etc. haben.

An Hochschulen sind das die Semester- und Diplomarbeiten, Berichte, Skripten oder Dissertationen. An Schulen sind dies Facharbeiten, Referate und ähnliches.

All diesen Dokumenten ist gemeinsam, daß sie eigentlich ganz einfach mit den Standardklassen `book`, `report` oder `article` bzw. den Pendanten aus dem KOMA-Script gut erstellt werden können. Für derartige Dokumente wurden ja schließlich diese L^AT_EX-Klassen ja auch ursprünglich geschaffen.

Eventuell müssen nur für das Titelblatt und die Gestaltung von Kopf- und Fußzeile geeignete Vorgaben gemacht werden. Dies kann durch `sty`-files relativ einfach und für eine Institution einheitlich erfolgen.

Oft besteht aber der Wunsch, das Titelblatt etwas gefällig zu gestalten. So soll z.B. ein Logo eingebunden werden oder es existieren andere Vorgaben für das Layout. Das umzusetzen, ist zwar für jemanden, der sich mit L^AT_EX gut auskennt kein allzu großes Problem; Anfänger sind aber damit oft überfordert.

Hier wäre es sehr nützlich, wenn man irgendwo – z.B. auf den `ctan` – einige Beispiele für derartige Titelblätter sammeln und als Basis für ein eigenes Layout zur Verfügung stellen könnte. Das wäre z.B. eine Aufgabe für den AK Schule.

3 Der leidige Papierkram

Jeder weiß, daß es neben den „großen“ Dokumenten auch noch sehr viel „kleine“ gibt, eben den sogenannten Papierkram. Das sind Briefe, Faxe, Memos, Aushänge, Protokolle, Aushänge, Bestätigungen und ähnliches. Meist sind das eher kurze Schriftstücke mit 1-2 Seiten, bei denen oft auch nicht so viel Wert auf Typografie und Gestaltung gelegt wird.

Diese Dokumente kann man entweder auch mit der Klasse `article` erstellen, oder man verwendet dafür spezialisierte Klassen, wie z.B. `scrletter` oder `din-brief`. Das Problem ist allerdings, daß man bei Briefen usw. oft konform gehen muß mit einem bereits vorhandenen Brief-Layout oder einer Corporate-Identity.

Natürlich schafft es ein L^AT_EX-Experte, etwa die `scrletter`-Klasse diesen Anforderungen anzupassen und so z.B. Adresse oder Logo einzubinden. Wünschenswert wäre auch hier wieder eine Sammlung von nützlichen Dokumenten, wie z.B.

- Fax,
- Kurznotiz,
- Protokoll.

4 Spezialitäten

Dann gibt es aber noch einige Dinge, die man weder mit den weitverbreiteten WYSIWIG-Systemen, noch mit L^AT_EX-Standardklassen vernünftig erstellen kann. Das sind z.B. Stundenpläne oder Prüfungsaufgaben oder Aufgabenblätter mit Musterlösungen.

4.1 Stundenpläne

Stundenpläne spielen an Fachhochschulen, Gymnasien und anderen Schulen eine große Rolle, bestimmen sie doch weitgehend den Tages- und Wochenablauf. Es gibt nur relativ

wenige vernünftige Lösungen für die Erstellung derartiger Dokumente.

4.1.1 Dokumentklasse `schedule` von Jason Alexander,

Für \LaTeX liegt zur Zeit das Paket `schedule`, von JASON ALEXANDER auf `ctan`. Es ist relativ leicht zu bedienen und liefert ansprechende Ergebnisse. Hier ein Beispiel mit `schedule`:

Stundenplan Prof. Lachenmayr SS 2001

	Monday	Tuesday	Wednesday	Thursday	Friday
8:00 am					
8:15 am					
8:30 am		WeKu1Me	MaschEl		
8:45 am		HT 2B A 408	HT 2B A 408		
9:00 am		14-tägig			
9:15 am					
9:30 am					
9:45 am					
10:00 am	MaschEl	MaschEl		WeKu1Me	
10:15 am	HT 2C A 410	HT 2A A 412		HT 2C A 408	
10:30 am					
10:45 am					
11:00 am				14-tägig	
11:15 am					
11:30 am					
11:45 am	WeKu1Me		MaschEl Übg		
12:00 am	HT 2A A 412		HT 2B A 413		
12:15 am			Gruppe 1		
12:30 am	14-tägig				
12:45 am					
1:00 pm					

Die Eingabe dazu sieht so aus:

```

1 \CellHeight{1cm}
  \CellWidth{3cm}
  \TimeRange{8:00-13:00}
  \SubUnits{15}
5 \BeginOn{Monday}
  \TextSize{\normalsize}
  \FiveDay

  \NewAppointment{uebung}{red}{black}
10 \NewAppointment{class}{cyan}{black}

  \begin{schedule}[Stundenplan Prof. Lachenmayr SS 2001]
    \class{\textbf{MaschEl}}{HT\,2C\quad A\,410}{M}{9:45-11:15}
    \class{\textbf{MaschEl}}{HT\,2A\quad A\,412}{T}{9:45-11:15}
15 \class{\textbf{MaschEl}}{HT\,2B\quad A\,408}{W}{8:00-9:30}
    \class{\textbf{WeKu1Me}}{HT\,2B\quad A\,408 \newline{\small 14-tägig}}{T}{8:00-9:30}
    \class{\textbf{WeKu1Me}}{HT\,2A\quad A\,412 \newline{\small 14-tägig}}{M}{11:30-13:00}
    \class{\textbf{WeKu1Me}}{HT\,2C\quad A\,408 \newline{\small 14-tägig}}{Th}{9:45-11:15}
    \uebung{\textbf{MaschEl Übg}}{HT\,2B\quad A\,413 \newline{Gruppe 1}}{W}{11:30-13:00}
20 \end{schedule}

```

Es benutzt die `picture`-Umgebung und das Paket `calc` um die Termine maßstäblich zu setzen. Durch Rückgriff auf das Paket `color` können die einzelnen Boxen farbig hinterlegt werden. Termine, die außerhalb des Zeitrahmens liegen werden ausgeblendet.

Der Nachteil dieses Pakets ist, daß es leider – wie viele andere Pakete aus dem angelsächsischen Raum – „hardcoded english“ ist. Es gibt keine Möglichkeit, um

- von der 12-Stunden Darstellung mit am/pm auf die 24-Stunden-Darstellung umzuschalten,

- die Namen der Tage auf deutsch umzustellen; sind ebenfalls englisch fest einprogrammiert und werden als Textstrings an vielen Stellen in den Makros direkt verwendet.

Für einen Anwender mit durchschnittlichen Programmierkenntnissen ist es daher schwierig, für einen Anfänger praktisch unmöglich, von diesem Makropaket eine deutschsprachige Variante zu erstellen.

4.1.2 Veraltete Dokumentklasse `schedule` von Frank Pfenning

Bis vor einiger Zeit war auf ctan das L^AT_EX2.09-Paket `schedule` von FRANK PFENNING (`fp@cs.cmu.edu`) verfügbar, das ich noch bis vor kurzen verwendet habe; es läuft problemlos im Kompatibilitätsmodus. Es benutzte ebenfalls die `picture`-Umgebung. Es liefert auch sehr gut brauchbare Pläne und ist weitgehend konfigurierbar, aber etwas umständlicher zu handhaben. Für eine Weiterentwicklung wäre evtl. auch dieses Paket eine Ausgangsbasis.

4.1.3 Tagungen usw.

Faßt man den Begriff Stundenplan weiter, so kommt man auf die Gestaltung von Programmen, wie etwa für diese Tagung. Während die alte Version von `scheudle` von Frank Pfenning (`fp@cs.cmu.edu`) in der Lage war, auch solche Dokumente zu erstellen, existiert meines Wissens z.Z. sonst keine geeignete Lösung dafür. Vielleicht könnte man die Makros mit denen wir das Tagungsprogramm für DANTE 2001 erzeugt haben, vernünftig dokumentiert, der Allgemeinheit zur Verfügung stellen.

4.2 Prüfungsaufgaben oder Aufgabenblätter mit Musterlösungen

Aufgabenblätter mit Musterlösungen sollen den Studenten bzw. Schülern helfen, ihr Wissen zu vertiefen, Prüfungen hingegen dienen zur Feststellung ihres Wissensstandes. Alte Prüfungen (mit Musterlösung) werden oft als Übungsaufgaben für die nachfolgenden Kurse ausgegeben.

Hier ist das Angebot an L^AT_EX-Klassen recht groß; es gibt:

`answers` von MIKE PIFF,

`exam` von PHILIP HIRSCHHORN,

`exams` (`exam`) von HANS VAN DER MEER

`examdesign` (`exam`) von JASON ALEXANDER.

4.2.1 Dokumentklasse `answers` von Mike Piff

Dieses Paket ist relativ einfach zu handhaben, wie ein mitgelieferte Beispieldatei zeigt:

```

1  %%
  %% This is file 'ansexam1.tex',
  %% generated with the docstrip utility.
  %%
5  %% The original source files were:
  %%
  %% answers.dtx (with options: 'ex1')
  %%

```

```

10 \documentclass[12pt,a4paper]{article}
   \usepackage{answers}
   \Newassociation{sol}{Solution}{ans}
   \newtheorem{ex}{Exercise}
   \begin{document}
15 \Opensolutionfile{ans}[ans1]
   \section{Problems}
   \begin{ex}
     First exercise
     \begin{sol}
       First solution.
20   \end{sol}
   \end{ex}
   \begin{ex}
     Second exercise
     \begin{sol}
25   \end{sol}
     Second solution.
   \end{ex}
   \Closesolutionfile{ans}
   \section{Solutions}
30 \input{ans1}
   \end{document}
   \endinput
   %%
   %% End of file 'ansexam1.tex'.

```

Es benutzt die Möglichkeit von \LaTeX , eine neue Umgebung zu definieren und die Lösungen in einer eigenen Datei zwischenspeichern, die dann wieder eingelesen wird, um das Lösungsblatt zu generieren.

Damit stellt es eine recht einfach zu handhabende Möglichkeit Aufgabenblätter oder Prüfungsaufgaben mit den dazugehörigen Lösungsblättern zu schreiben.

Der große Vorteil ist, daß nur auf Standard- \LaTeX -Befehle zurückgegriffen wird und keine komplexen eigenen Makros definiert werden. Dadurch sind keine Unverträglichkeiten mit anderen Paketen zu erwarten. Der Benutzer kann die Dokumente mit seinen gewohnten Paketen gestalten.

4.2.2 Dokumentklasse `exam` von Philip Hirschhorn

Auch dieses Paket benutzt weitgehend die Standard- \LaTeX -Befehle bzw. die Möglichkeiten neue Umgebungen usw. zu definieren. Das Paket kommt mit einer sehr ausführlichen Dokumentation, die die Benutzung sehr erleichtert. Der Autor schreibt selbst

Das meiste, was die Dokumentklasse `exam` bietet, kann auch durch Verwendung von `fancyheadings.sty`, Anpassen der Seiten-Layout-Parameter und sorgfältige Verwendung der `list`-Umgebung erreicht werden, aber die Dokumentklasse `exam` versucht, dies so weit wie möglich zu vereinfachen.

So können z.B. die Seiten Kopf *und* Fuß enthalten, auch mehrzeilig. Das zentrale Element ist die `question`-Umgebung, die zur weiteren Untergliederung noch `parts` und `subparts` haben kann. Mit `uplevel` oder `fullwidth` können Bemerkungen oder erläuternder Text mit reduziertem Einzug auch zwischen den einzelnen Fragen positioniert werden.

Eine Besonderheit ist die Möglichkeit, die Punkte, die eine Frage wert ist, gleich mit anzugeben. `exam` ermöglicht entweder die Positionierung am Rand oder im Text, auch in

einem Rahmen. Außerdem werden die Punkte durch `\addpoints` automatisch zusammengezählt und stehen über das Makro `\numpoints` zur Verfügung. Ähnliches gilt für die Zahl der Fragen oder die Gesamtzahl der Seiten.

Sehr weitgehend sind die Möglichkeiten, Kopf und Fuß anzupassen; sie sind in der Dokumentation bestens erläutert, so daß an dieser Stelle nicht näher darauf eingegangen werden braucht.

Was diese Dokumentklasse allerdings *nicht* bietet, ist die automatische Verwaltung der zu den Fragen gehörigen Musterlösungen. Diese müssen also separat auf konventionellem Weg erzeugt werden.

Insgesamt macht dieses Paket einen recht ausgereiften und überlegten Eindruck; dies zeigen Details, daß man z.B. mit dem Kommando `\pointsname` einen beliebigen Text, also z.B. „Punkte“ statt der Voreinstellung „points“ eingeben kann. So kann das Paket problemlos an die Bedürfnisse einer deutschen (Hoch-) Schule angepaßt werden.

4.2.3 Dokumentklasse `exam` von Hans van der Meer (`exams`)

Die Dokumentklasse `exam` von HANS VAN DER MEER ist wegen der Namensgleichheit auf `ctan` unter dem Verzeichnis `exams` zu finden. In ihrem Leistungsumfang geht sie deutlich weiter als die zuvor diskutierten Pakete. Einige Besonderheiten sind:

- ein Dokument kann *mehrere* Prüfungen beinhalten, die jeweils durch `\begin{exam}` ... `\end{exam}` erstellt werden (ähnlich wie bei der `letter`-Klasse). Somit kann man ganz einfach mehrere Varianten einer Prüfung in einer Datei halten oder Prüfungssammlungen herausgeben.
- Man kann den Text *parametrieren*, um z.B. für mehrere Gruppen die gleiche Frage, nur mit anderen Worten oder Zahlenwerten stellen. Dabei können diese Parameter durch Randomisierung auf Wunsch noch beliebig gemischt werden.
- Den Fragen können mit dem Kommando `\score` Punkte zugeordnet werden, die auch automatisch aufsummiert werden. In der Musterlösung werden diese Punkte gedruckt, in der Prüfung erscheint ein leeres Kästchen für die Korrektur.
- Den Fragen können Antworten zugeordnet werden. Durch die Option `answers` werden die Lösungen mit erzeugt.
- Das Paket kann über die Konfigurationsdatei `exam.cfg` relativ einfach an die eigenen Bedürfnisse angepaßt werden.

Die Grundidee ist, alle Fragen in jeweils eigenen Dateien zu halten. In diesen Fragen stehen auch die jeweilige Lösungen. Diese Dateien werden dann mit dem Kommando `\question` eingebunden. Verzeichnisse (z.B. für Bilder oder häufig auftretende, gemeinsam verwendete Dateien) können voreingestellt werden. Das ist idela für Situationen, in denen eine Art „Datenbank“ aus eher kurzen Fragen aufgebaut wird.

In den Fragedateien können drei Typen von Aufgaben stehen:

1. `problem` für normale Fragen,
2. `parameterproblem` für Fragen, bei denen Teile des Textes parameteriert werden

3. choice für Multiple-Choice-Fragen.

Ein Beispiel für eine einfache Eingabedatei ist:

```
1  \documentclass[answers,titlepage]{exam}
   \documentclass[titlepage]{exam}

   \usepackage{german}
5  %\usepackage[T1]{fontenc}
   \usepackage{ae}
   \usepackage{units}
   \usepackage[ansinew]{inputenc}
   \usepackage{latexcad}
10  \usepackage[final]{graphicx}

   \begin{document}
   \title{Prüfung Energietechnik}
   \author{Prof. Dr. Georg Lachenmayr}
15  \target{Studiengang Holztechnik 4. Semester}
   \maketitle
   \begin{exam}[12354]{20.01.2001}
   \textsl{\fullanswer}
   \section{Druckluft}
20  Ein trockenlaufender Schraubenkompressor mit  $\text{\unit[28]{\nicefrac{m^3}{min}}}$ 
   Ansaugvolumen erzeugt Druckluft von
    $\text{\unit[10]{bar}}$  Überdruck und einer Temperatur von  $\text{\unit[20]{^\circ C}}$  aus Luft von
    $\text{\unit[20]{^\circ C}}$ . Direkt nach der Verdichtung hat die Druckluft eine Temperatur von
    $\text{\unit[185]{^\circ C}}$ . Der Motor des Kompressors nimmt eine elektrische Leistung
25   $P_{\text{el}} = \text{\unit[200]{kW}}$  auf.

   \vspace*{5mm}

   \question{}{FrageDL1}
30  \question{}{FrageDL2}
   \question{}{FrageDL3}
   \end{exam}
   \end{document}
```

Die erste Fragedatei `FrageDL1.tex` ist:

```
1  \begin{problem}Wie groß ist der Polytropenexponent?
   \score{2}
   \begin{longanswer}
   Aus Tabelle 4.1:
5  
$$n = \frac{1}{\ln\frac{11}{1}} - \frac{1}{\ln\frac{273+185}{273+20}} = 1,229 \approx 1,23$$

   \end{longanswer}
   \end{problem}
   \endinput
```

Jeder Aufruf von `\question` erzeugt eine Frage mit einer Nummer; die Numerierung läuft durch die ganze Prüfung, d.h. durch eine ganze `exam`-Umgebung.

Die Klasse `exam` von HANS VAN DER MEER ist sehr gut und ausführlich dokumentiert. Sie ist verträglich mit den Standard- \LaTeX -Befehlen, so daß die üblichen Befehle für Abschnitte oder Listen verwendet werden können. Durch die weitgehende Konfigurierbarkeit läßt sich das Paket sehr leicht auf deutsche Verhältnisse anpassen.

Allerdings läßt das Paket auch noch *einige Wünsche* offen. Für mich als Nachteil ist zu bewerten, daß die Fragen *keine Unterpunkte* haben können. Dazu kommt noch, daß jede (Teil-) Frage in einer *eigenen Datei* stehen muß. Eine typische Energietechnik-Klausur hat 5–6 Hauptfragen mit jeweils 3–6 Unterfragen; das gibt dann zusammen 30 Dateien,

was etwas lästig ist. Besser und übersichtlicher wäre es, wenn die Hauptfragen mit ihren Unterfragen zusammen in einer Datei stehen könnten.

4.2.4 Dokumentklasse exam von Jason Alexander (examdesign)

Auch diese Dokumentklasse hat ursprünglich den Namen `exam` und ist deshalb wegen Namensgleichheit auf `ctan` unter `examdesign` zu finden. Auch Sie ermöglicht die Bindung von Frage und Antwort, Parametrierung des Textes und Randomisierung, d.h. das automatische Durchmischen von Fragen für mehrere Varianten ein und derselben Prüfung.

Es gibt folgende Typen von Fragen, die als Umgebungen definiert sind:

`fillin` Lückentext, in dem der Prüfling Worte ergänzen muß.

`shortanswer` für „normale“ Fragen, die entweder mit einem Satz oder einer Rechnung zu beantworten sind..

`truefalse` fragen, die mit „Ja“ oder „Nein“ zu beantworten sind.

`multiplechoice` Multiple-Choice-Fragen zum Ankreuzen.

`fixed` für „normale“ Fragen; ist im Prinzip das gleiche, wie `shortanswer`; der Unterschied ist nur, daß bei Randomisierung dies Frage *nicht* wandert.

Die Anwendung ist recht einfach, wie folgendes Fragment zeigt; das optionale Argument `[!]` von `\choice` zeigt die richtige Lösung an, was in der Musterlösung dann entsprechend hervorgehoben wird.

```
1 \begin{multiplechoice}
  \begin{question}
    How much wood could a woodchuck chuck if a woodchuck could chuck wood?
    \choice{A lot.}
5   \choice{More than most.}
    \choice{Exactly  $\pi$  cords.}
    \choice[!]{It depends on the nature of the woodchuck.}
  \end{question}
\end{multiplechoice}
```

Ein anderes Beispiel ist im folgenden Fragment zu sehen, es entspricht dem aus Abschnitt 4.2.3:

```
1 \begin{fixed}[{\Large Druckluft}]{\large\fbbox{6 Punkte}}]
  \noindent
  Ein trockenlaufender Schraubenkompressor mit  $\frac{m^3}{\text{min}}$ 
  Ansaugvolumen erzeugt Druckluft von
5 \unit[10]{bar} Überdruck und einer Temperatur von  $^{\circ}\text{C}$  aus Luft von
   $^{\circ}\text{C}$ . Direkt nach der Verdichtung hat die Druckluft eine Temperatur von
   $^{\circ}\text{C}$ . Der Motor des Kompressors nimmt eine elektrische Leistung
   $P_{\text{el}}$  auf.

10 \begin{question}Wie groß ist der Polytropenexponent?
  \newline\mbox{\hfill\fbbox{2 Punkte}}
  \examvspace*{2cm}
  \begin{answer}
  Aus Tabelle 4.1:
15 
$$n = \frac{11}{1} - \frac{273+185}{273+20} = 1,229 \approx 1,23$$

  \end{answer}
```

```

\end{question}
20 \begin{question}Wie gro"s ist der Wirkungsgrad  $\eta$ ? \hfill\fbbox{2 Punkte}
\examvspace*{2cm}
\begin{answer}
\dots
\end{answer}
25 \end{question}

\begin{question} Wie hoch \dots
\begin{answer}
\dots
30 \end{answer}
\end{question}

\end{fixed}

```

Man erkennt die Besonderheit, daß die `question`-Umgebung die Antwort, also die `answer`-Umgebung *umschließt*.

Die Klasse bietet zahlreiche Hilfen zur Formatierung an, wie z.B. das Kommando `\examvspace`, mit dem man Platz für die schriftliche Lösung freihalten kann. In der Musterlösung (`key`) wird dieser Platz natürlich weggelassen. Man kann auch ganze Textbereiche so auszeichnen, daß sie nur in Prüfung oder Musterlösung erscheinen; auch der Kopf und Schluß des Dokuments kann für beide Fälle separat gestaltet werden.

Durch das Kommando `\word` ist es möglich parametrisierten Text einzufügen. So sieht z.B. `P=\word{200}{220}{300}\unit{kW}` drei Varianten einer Leistung vor. Wenn am Anfang des Dokuments der Schalter `\NumberOfVersions{3}` gesetzt wird, dann erzeugt `exam` automatisch drei Prüfungen, in denen jeweils ein anderer Wert für die Leistung P steht.

Zusätzlich bietet die Klasse `exam` von JASON ALEXANDER die Möglichkeit, daß für mehrere Gruppen die verschiedenen Fragen, also die Environments automatisch zufällig vertauscht werden. So erhalten die Prüflinge Prüfungen, die zwar den gleichen Inhalt haben, aber auf den ersten Blick unterschiedlich aussehen.

Das Paket ist ebenfalls sehr ausführlich dokumentiert und hat sich bei mir jetzt schon seit mehreren Semestern bewährt. Es läßt auch nur wenige Wünsche offen.

Seine Nachteile sind:

1. Aufgrund der Komplexität dieser Klasse und des Programmierstils gibt es *Einschränkungen*, weil einige \LaTeX -Kommandos undefiniert werden. So funktionieren z.B. das `\verb`-Kommando und die `enumerate`-Umgebung *nicht* in der `exam`-Klasse.
2. Es gibt keine Möglichkeit, die *Punkte* aufzusummieren; daher die „handgeschnittzte“ Lösung im letzten Beispiel.
3. Besonders lästig ist, daß die einzelnen Environments die Überschrift „Section 1.“ usw. erhalten und nicht etwa „Frage 1.“; das ist einfach unschön. Ein Blick in den Code der Klasse zeigt aber, daß leider diese Bezeichnung auf deutsch umzustellen recht aufwendig ist, weil das englische Wort fest einprogrammiert ist und als Textstring an vielen Stellen in verschiedenen Makros direkt verwendet wird. Für einen Anwender mit durchschnittlichen Programmierkenntnissen ist es schwierig, für einen Anfänger praktisch unmöglich, von diesem Makropaket eine deutschsprachige Variante zu erstellen.

4.2.5 Fazit

Es gibt derzeit keine Klasse, für Prüfungsaufgaben, die wirklich alle Wünsche erfüllt, die man aus der Praxis stellt. Hier wären \TeX -Spezialisten gefragt, die in der Lage wären, die guten Ansätze von zwei oder drei der vorgenannten Pakete zu vereinigen.

5 Jenseits vom Papier

Die Zeiten sind vorbei, in denen bedrucktes Papier das Höchste der Gefühle war. Wichtig sind folgende Bereiche:

5.1 OHP-Projektion

Die OHP-Projektion ist heute Standard und wird gerade an der Hochschule im lehrerzentrierten Unterricht ergänzend zum oder anstelle der Tafel eingesetzt. Dabei wird auch im Zeitalter von Notebook und Beamer die gute alte OHP-Folie noch sehr häufig eingesetzt. Sie erlaubt z.B. auf einfache Weise handschriftliche Ergänzungen. Außerdem sind nicht in jedem Hörsaal PC und Beamer verfügbar, wohl aber ein OHP; daher ist der Aufwand, Folien einzusetzen geringer.

Daher kommt der Erstellung von Folien für die OHP-Projektion immer noch eine wichtige Rolle zu. Hat man eine Datei mit mehreren derartigen Folien erzeugt, dann kann man sie auch mit `gsview`, `acrobat` oder ähnlichem auch als einfache OHP-Präsentation mit dem Beamer verwenden.

An dieser Stelle möchte ich auf eine Problematik hinweisen, die gerne übersehen wird: Bei größeren Vorlesungen hat typischerweise ein Skriptum 200–300 Seiten, die dazu gehörigen OHP-Folien füllen (in Hüllen) 2-3 Ordner. Es ist *äußerst* lästig und *fehlerträchtig*, wenn man nun bei *Änderungen am Skriptum* die entsprechenden Stellen in den *Folien von Hand ändern* muß.

Sicher kann man die geänderten \LaTeX -Konstrukte in die Folien-Datei kopieren, besser wäre es aber, wenn man *nur eine Mutterdatei* hätte, von der aus man *wahlweise* den gedruckten Text (Skriptum) *oder* die OHP-Folien erzeugen kann.

Meines Wissens gibt es dazu zur Zeit noch keine Lösung, die wirklich durchgängig funktioniert. Aber ich denke, daß es gerade ein Stärke von \LaTeX -Dokumenten ist, durch *logisches Mark-Up* dieses Problem weitgehend automatisch zu lösen.

Nachdem ich früher sehr viel mit dem `slides`-Style gearbeitet habe, verwende ich jetzt seit einiger Zeit die Dokumentklasse `seminar`. Sie erspart zwar Handarbeit nicht völlig, reduziert sie aber erheblich.

Sicherlich könnte man mit anderen Präsentationspaketen (PP4 etc.) oder pdf\TeX hier andere Ansätze verwirklichen; das Hauptproblem – so denke ich – ist aber, das *logische Mark-Up*, das steuert, was ins Skriptum oder auf die Folien soll, auch wirklich konsequent anzuwenden.

5.2 Web-Auftritte und andere elektronische Medien

In Zukunft stark zunehmen wird auch der Wunsch, neben der gedruckten Ausführung der Dokumente auch gleichzeitig elektronische Varianten zu erstellen, die man z.B. ins Web oder Intranet stellen kann.

So ist es bei uns schon seit einiger Zeit üblich, Skripten, alte Prüfungsaufgaben und Lösungen auf unseren Novell-Servern den Studenten zum Download anzubieten. Hier kann insbesondere durch Konvertierung der Dokumente nach pdf sehr viel Plattenspeicherplatz und Downloadzeit gespart werden, abgesehen von dem Ärger, den man mit der Inkompatibilität der verschiedenen Versionen von W... hat.

Andere elektronische Medien spielen zur Zeit bei uns noch kaum eine Rolle. Ich könnte mir aber vorstellen, daß man z.B. anstelle eines Skriptums eine Art Lern-CD anbietet, die neben dem Text, der dann aber mit Hyperlinks gut erschlossen sein muß, auch noch Übungsaufgaben, Rechenblätter oder animierte Demos enthält.

Sicherlich könnte so etwas auch im Web verfügbar gemacht werden. Aber dann hat man überhaupt keine Kontrolle mehr, wer irgend etwas herunterlädt – und das noch kostenlos! Daher wird man sicherlich die Dokumente, die ins Web gestellt werden, sehr sorgfältig auswählen.

Derartige Dinge sind in Entwicklung, wie die Projekte „Virtuelle Hochschule Bayern“ oder „MeiLe“, (Medieneinsatz in der Lehre), die hier vom Wissenschaftsministerium aktuell abgewickelt werden.

6 URL der besprochenen Pakete:

- <ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/schedule/>
- <ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/answers/>
- <ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/exam/>
- <ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/exams/>
- <ftp://ftp.dante.de/tex-archive/macros/latex/contrib/supported/examdesign/>