

---

## T<sub>E</sub>X People: The TUG interviews project and book

Karl Berry and David Walden

### Abstract

We present the history and evolution of the TUG interviews project. We discuss the interviewing process as well as our methods for creating web pages and a printed book from the interviews, using m4 as a preprocessor targeting either HTML or L<sup>A</sup>T<sub>E</sub>X. We also describe some business decisions relating to the book. We don't claim great generality for what we have done, but we hope some of our experience will be educational.

### 1 Introduction

Dave had two ideas in mind when he suggested (originally to Karl) this interview series in 2004: (a) technology is created by and evolves with use by people, and the points of view and backgrounds of the people influence the technology; (b) there are lots of people (such as himself) who are relatively new to the T<sub>E</sub>X community, who therefore do not know much about the people who are already significant contributors to the community, and who may be curious to know more about past and current contributors to the T<sub>E</sub>X community. We also hoped that even long-time members of the T<sub>E</sub>X community would enjoy reading the interviews and learn things they did not previously know about their fellow T<sub>E</sub>X people.

Dave got the idea for an interview series from reading the books *Mathematical People* (edited by Donald J. Albers and Gerald L. Alexanderson, Birkhauser, 1985) and *More Mathematical People* (edited by Albers, Alexanderson, and Constance Reid, Harcourt, Brace, Jovanovich, 1990). The first of these collections of interviews of mathematicians includes a wonderful interview of T<sub>E</sub>X creator Donald Knuth (posted at [tug.org/interviews/interview-files/birkhauser-knuth](http://tug.org/interviews/interview-files/birkhauser-knuth), by permission from Birkhauser).

Dave's first concern was whether anyone would want to do an interview with him, an unknown in the T<sub>E</sub>X community. He was gratified that Dan Luecking (whose answers on `comp.text.tex` he admired) and Lance Carnes (with whom he was working on *The PracT<sub>E</sub>X Journal*, published at [tug.org/pracjourn](http://tug.org/pracjourn)) agreed to be the first interview subjects. These interviews provided a model to which he could point the next people he asked to do interviews, while also liberally dropping Karl's name. In time, the interview series seemed to gain advance acceptance with potential interview subjects.

### 2 The interview method

Interview subjects are<sup>1</sup> chosen based on (a) seeking diversity in many dimensions, (b) recommendations from people about who should be interviewed, and (c) potential interview subjects being willing to be interviewed.

The interviews have almost all been done via exchange of email using plain text; a couple of exceptions were done in L<sup>A</sup>T<sub>E</sub>X.

For the first several interviews, Dave sent a more or less complete list of questions (once the interviewee agreed to participate), on the theory that this would minimize the burden on the interviewee. However, having a dozen questions to answer at one time proved to be daunting to interviewees. Thus, Dave switched to a process wherein he sent a couple of standard initial questions (namely, "Tell me a bit about yourself and your history outside of T<sub>E</sub>X," and "How and when did you first get involved with T<sub>E</sub>X?"). The answers to early questions then guide a couple of follow-on groups of questions.

The interviewees are encouraged to answer spontaneously and at any length they desire, both to ease the burden of answering the questions and to spur spontaneity. Once enough questions have been asked and answered, Dave converts the plain text into HTML, possibly reordering some of the questions and answers to improve the flow, does other little bits of necessary editing (all the while trying to maintain the voice of the interviewee), and submits the near final draft to the interviewee for review and any desired changes.

Karl proofreads every interview before it was posted. He also frequently suggests additional questions, provides contact information, and generally encourages and supports Dave in his interview efforts. The interviews are then posted on the TUG web site: [tug.org/interviews](http://tug.org/interviews).

The average interview takes a few weeks of elapsed time. The shortest interview took only a few days. The longest interview took over a year. In a few cases, Dave has interviewed two people at one time — people who were well known for working closely together. In one case, Dave shared the interviewing duties with another person (the interview of Frank Mittelbach, which was also published in the *Free Software Magazine*. In another case (the interview of Raph Levien) Karl shared interviewing duties with another person. All the rest have been one-on-one interviews.

For quite a while Dave manually reformatted

---

<sup>1</sup> The interview series remains an ongoing process, so we describe it here in the present tense.

the interviews in HTML from the original plain text questions and answers as part of his editing pass. He is biased toward basic tools (WinEdt, Emacs, FTP, etc.) and does not use a “web publishing system”.

For the first interviews, we did not think about photos. Then we decided to include photos, began asking for them from current interviewees and also asked the early interviewees for photos. We slipped up here, in light of our later decision to create a book version of the interviews: we asked for photographs with only enough resolution for web display and not enough for printing in a book. So we later had to go back to most interviewees again for higher-resolution images.

### 3 The idea for the book

After a while, we began to discuss the possibility of eventually producing a book collecting the interviews, after we had about three years worth, with a dozen or so interviews done a year. Consequently, in our interview requests, we also started asking the interviewee to agree for his or her interview to be transcribed from HTML to L<sup>A</sup>T<sub>E</sub>X and be included in the book. We also confirmed that this was ok with earlier interviewees.

A little while later, Dave decided to create a small set of m4 macros ([gnu.org/software/m4](http://gnu.org/software/m4)) in terms of which he would transcribe the plain text interviews, and the file with the m4 macros would be compiled into HTML for the interviews. His idea was that later another set of m4 macros could convert the same m4 file to L<sup>A</sup>T<sub>E</sub>X for the book. All these definitions were in a file called `htmldefs.m4`, and an interview was compiled from its m4 file with a command such as

```
m4 htmldefs.m4 SomeInterviewee.m4 \  
> SomeInterviewee.html
```

An abridged set of the initial m4-to-HTML definitions is listed in Table 1. Here are a couple of specific examples:<sup>2</sup>

```
define([[_par]], [[<p>]])      (paragraph)  
define([[_it]], [[<i>${1}</i>]]) (italics)
```

To generate L<sup>A</sup>T<sub>E</sub>X instead of HTML, we made alternate definitions which output `\par` and `\textit{${1}}`.

The individual HTML interviews, both before and after using m4 as a stepping stone to HTML, had links in alphabetical order from the interviews home page which Dave created manually on the TUG server with Emacs. He also manually created the HTML

**Table 1:** Some definitions and their purposes from an early version of our `htmldefs.m4` file.

<code>_interviewee</code>	used in header and first question
<code>_intervieweeinitials</code>	used in following questions
<code>_interviewer</code>	used in first question
<code>_interviewerinitials</code>	used in following questions
<code>_par</code>	start paragraph
<code>_header</code>	takes photo file and blurb as parameters
<code>_question1</code>	formatting of first question
<code>_answer1</code>	formatting of first answer
<code>_question</code>	formatting of following questions
<code>_answer</code>	formatting of following answers
<code>_footer</code>	date, etc., at end of interviews
<code>_link</code>	link given text to a URL
<code>_url</code>	print and link given URL
<code>_it</code>	italics
<code>_emph</code>	also italics
<code>_ti</code>	italics within italics
<code>_tt</code>	typewriter
<code>_anglebrackets</code>	print argument between <code>&lt;</code> and <code>&gt;</code>
<code>_Dash</code>	em-dash
<code>_dash</code>	en-dash
<code>_amp</code>	ampersand
<code>_quote</code>	double-quote argument
<code>_verbatimstart</code>	start verbatim block
<code>_verbatimstop</code>	end verbatim block
<code>_verb</code>	inline verbatim
<code>_orderedliststart</code>	start ordered list
<code>_orderedliststop</code>	end ordered list
<code>_unorderedliststart</code>	start unordered list
<code>_unorderedliststop</code>	end unordered list
<code>_item</code>	begin an item in any list
<code>_TeX, ...</code>	lots of logos which are mostly plain text in HTML, but something fancier in T <sub>E</sub> X
<code>_Schopf, ...</code>	people’s names with diacritics
<code>_Ecole, ...</code>	other words with diacritics

file listing the interviews in chronological order. The result of all was posted at [tug.org/interviews](http://tug.org/interviews).

Some readers may wonder why we didn’t use a more au courant solution, such as creating the interviews in XML and arranging for that to generate HTML initially and L<sup>A</sup>T<sub>E</sub>X later, or doing the interviews in L<sup>A</sup>T<sub>E</sub>X (which could be used for the book) and using one of the L<sup>A</sup>T<sub>E</sub>X-to-HTML converters for the web site. The answer is simple: Dave and Karl already knew something about m4 and saw nothing to be gained by struggling with anything new (and arguably more complex).

### 4 Going ahead with the book

In late 2008 we made the decision to develop the book. We anticipated self-publishing it using print-on-demand from a company like Lightning Source

<sup>2</sup> Readers can mentally skip over the `[[ and ]]` constructs, which we use as the m4 “quoting” characters, allowing arguments to contain characters special to m4 such as commas and parentheses. More about this later.

(LSI, [lightningsource.com](http://lightningsource.com)). Karl chose  $7 \times 10$  inches as the page size from the sizes supported by LSI, because it was not so big that it would require a two-column format, but otherwise the biggest available, so minimizing the number of pages and print cost.

Karl developed the m4-to-L<sup>A</sup>T<sub>E</sub>X definition file ([tug.org/interviews/book/texdefs.m4](http://tug.org/interviews/book/texdefs.m4)) for converting the m4 files for each interview into L<sup>A</sup>T<sub>E</sub>X files for each interviews. Naturally, this evolved as new issues were discovered as the conversion into L<sup>A</sup>T<sub>E</sub>X proceeded. As an example, one of the shorter m4 interview files is available at [tug.org/interviews/book/gordon.m4](http://tug.org/interviews/book/gordon.m4) (Peter Gordon's interview) and the resulting L<sup>A</sup>T<sub>E</sub>X file at [tug.org/interviews/book/gordon.inc](http://tug.org/interviews/book/gordon.inc).

Some of the interviews had never been in the m4 format, and Karl converted the HTML for these earlier interviews into the m4 format using Emacs, Perl, and other Unix tools. This enabled all the interviews to be handled uniformly for the book and provided the option of later converting all the interviews, as cleaned up for the book, back to HTML to improve the interviews web site. Karl then edited all of these m4 files further to eliminate typos, other broken or awkward text, and generate well-typeset L<sup>A</sup>T<sub>E</sub>X, following the same general conventions long-established for *TUGboat*.

In addition to Karl, Barbara Beeton (whose reputation for editing prowess is well known in the T<sub>E</sub>X community) also edited each interview, reviewed the formatting, and so on. As with *TUGboat*, Barbara and Karl strived to achieve appropriate English for each interview while maintaining as much as possible the interviewee's voice.

Next, Dave circulated the edited and typeset interviews back to the interviewees for approval and possibly for an additional update, usually written as an endnote; most interviewees chose to leave their interview current as of the original interview end date. Dave also began to solicit higher resolution photos as part of these exchanges with the interviewees (since we desired the photos to be about 2 inches wide, we asked for photos that were at least 600 pixels wide); Dave also used Photoshop to convert many of the color photos to grayscale and to adjust the contrast, etc., of all photos to improve how well they printed in a black and white book.

Finally, Dave and Karl drafted the preface, title pages, and other front matter, and integrated them into the Makefile from which the entire book was compiled.

## 5 Technical L<sup>A</sup>T<sub>E</sub>X details

In addition to the `texdefs.m4` file mentioned above, Karl developed a Makefile ([tug.org/interviews/book/GNUMakefile](http://tug.org/interviews/book/GNUMakefile)) to automatically convert all of the m4 files into L<sup>A</sup>T<sub>E</sub>X files, and then compile the L<sup>A</sup>T<sub>E</sub>X files for the interviews and the frontmatter files into one complete PDF. The master L<sup>A</sup>T<sub>E</sub>X file which the Makefile calls to compile the book is at [tug.org/interviews/book/ivbook.tex](http://tug.org/interviews/book/ivbook.tex).

### 5.1 Table of contents

We considered several orderings for the interviews: alphabetical was an obvious possibility, but added nothing over a simple list of names; another was grouping by category, such as putting related developers together, but this proved too vague to be meaningful. Finally we settled on chronological order.

We decided to have two tables of contents, one chronological (the normal page number order, in this scheme, written by L<sup>A</sup>T<sub>E</sub>X), and one alphabetical. Karl wrote a script to sort the names and generate the alphabetical contents from the chronological one ([tug.org/interviews/book/a-lphabetize](http://tug.org/interviews/book/a-lphabetize)).

### 5.2 The book style file

The master file, `ivbook.tex`, is mostly just a list of the frontmatter and interview files in the order we wanted to print them; the only notable thing is the initial setup:

```
\documentclass{book}
\usepackage{ivbook}
```

That is, we use the basic L<sup>A</sup>T<sub>E</sub>X book style, augmented with a style file. We have found that collecting all customizations and settings in such a central style file is a good idea for a significant projects.

Here are the main things which our book style file does ([tug.org/interviews/book/ivbook.sty](http://tug.org/interviews/book/ivbook.sty)):

- It loads some standard packages: `geometry` to specify the page layout, `graphicx` for handling the photos, `microtype` to help with line breaking, `url` for line breaking on urls, and `fancyhdr` to specify our desired running header (and an empty running footer).
- It loads `babel` with support for Vietnamese, for the sake of two so-called “horned” letters in the interview with Philip Taylor. We were grateful that Hàn Thế Thành had created Vietnamese glyphs for our font (among many others) as part of the VnT<sub>E</sub>X project ([vntex.sf.net](http://vntex.sf.net)).
- It specifies the fonts used. Karl wanted to use a freely available font that was included in T<sub>E</sub>X Live, and looked through the online L<sup>A</sup>T<sub>E</sub>X Font

Catalogue ([tug.dk/FontCatalogue](http://tug.dk/FontCatalogue)). He then experimented with different choices—the Font Catalogue helpfully lists the L<sup>A</sup>T<sub>E</sub>X commands to use, making such experimentation easy—and settled on Charter, as extended by the Math Design Project:

```
\usepackage[bitstream-charter]{mathdesign}
```

Unfortunately, the last release of the free Charter from Bitstream (used by Math Design) includes a number of unfortunate kerning pairs, especially with punctuation, such as ‘P.’. Karl ended up remaking the TFM for the main text font with the kerning from an older release (the `bchr8t.tfm` file which is in T<sub>E</sub>X Live). He wrote an ugly script to get this job done ([tug.org/interviews/book/kernfix](http://tug.org/interviews/book/kernfix)).

For the typewriter material, we used the Inconsolata font by Raph Levien, one of the interviewees. Karl wrote the L<sup>A</sup>T<sub>E</sub>X support macros, and later released them to CTAN ([mirror.ctan.org/fonts/inconsolata](http://mirror.ctan.org/fonts/inconsolata)) and T<sub>E</sub>X Live.

- Because the interviews are each only a few pages long, the style file eliminates blank verso pages by redefining `\cleardoublepage`, as suggested in the FAQ ([www.tex.ac.uk/faq](http://www.tex.ac.uk/faq)).
- It defines a macro `\ivchapter` to make each interview into a separate (unnumbered) chapter, and add the name to the table of contents.
- Speaking of the tables of contents, it also makes various (re)definitions for formatting those; one that may be of interest is setting `\@pnumwidth`, the page number width, to `2.5em`; the default of `1.55em` is not enough for three digits in Charter, unlike Computer Modern.
- It defines a macro `\ivblurb` to format the beginning of each interview, including the brief description and photograph. Some of this is pretty intricate; the interested reader may like to study the code (due to Barbara Beeton) which makes the “[Interview completed . . .]” text be typeset flush right on the last line if it fits, or on its own line if not.
- It defines macros with which to format the questions and answers.
- It defines common macros `\Dash` for formatting dashes (we like thinspaces around them), `\acro` for acronyms (we like to set “words” in all caps one point size smaller than the main text), etc. We adopted these and many other conventions from *TUGboat*.
- It defines a lot of logos, both simple and complex. Because we’re not using the Computer Modern

fonts, we used the alternative logo definitions for L<sup>A</sup>T<sub>E</sub>X and the like developed by Grzegorz Murzynowski, as written up in *TUGboat* 29:1 (EuroT<sub>E</sub>X 2007 proceedings).

- It changes the percent sign and dollar sign into normal characters (`\catcode 12`), since they come up from time to time in the interviews, and enables DEL (aka `^^?`) as a comment character.
- It handles indenting of verbatim blocks. Karl was surprised to learn that `verbatim` mode operates by typesetting a one-item list without any list marker. This meant the usual mechanisms for indentation, such as `\leftskip`, were rendered ineffective. We ultimately resorted to loading `fancyvrb` and using its `xleftmargin` setting.
- Finally, we wanted to specify the vertical space above lists. This turned out to be surprisingly difficult. The L<sup>A</sup>T<sub>E</sub>X parameter documented to control this is `\topsep`. However, L<sup>A</sup>T<sub>E</sub>X saves a copy of the original list parameters in the macro `\@listI` (source file `classes.dtx`), which is invoked after every font size change. Our document, like most documents, changes font size from time to time (for the heading beginning each interview, etc.). Thus, to affect the spacing above lists once and for all, it is necessary to manually redefine `\@listI`, and not just set `\topsep`.

## 6 Technical m4 details

We ended up being reasonably happy with our choice of m4 as the “high-level” language in which to write the interviews. Almost all our definitions are trivial text expansions that took no time to write and served the purpose of making it possible to generate either HTML or L<sup>A</sup>T<sub>E</sub>X without much trouble. Below we discuss two of the nontrivial definitions. (Incidentally, although all our m4 commands start with `_`, this is not a requirement of the language.)

### 6.1 Converting date formats

One nontrivial definition related to handling the interview dates. In the interview sources, we specify the dates like this, for an interview completed on June 25, 2005:

```
define([[_date]], [[2007-06-25]])
```

(As mentioned in a previous footnote, the `[[. . .]]` construct is our way of quoting, stopping m4 from expanding the quoted text further.) In the printed book, though, we wanted to show the date like this:

Interview completed 25 June 2005.

How to convert from one date format to the other? We didn't want to just change the source since the numeric format is used online, and besides, that would have been extra work. We ended up with these m4 definitions to do the job:

```
define([[_header]],dnl
[[... completed _ivdate(_date)]]dnl
define([[_ivdate]],dnl
[[esyscmd(date +%e %B %Y' -d '$1')]]dnl
```

Explanation:

- `dnl` stands for “discard-to-newline”, i.e., starts a comment. We use it here because there's no point in passing the newlines after each definition to the L<sup>A</sup>T<sub>E</sub>X output file.
- We use a helper macro `_ivdate` to do the transformation, and pass it the original (numeric) date from the source file.
- `_ivdate` uses the m4 builtin function `esyscmd` (“execute system command”) to call the GNU date utility, which recognizes the numeric source date and reformats it in the way we want. The result is inserted in the normal output text; `esyscmd`, and therefore `_ivdate`, expands to the result of running the command, like ‘...’ in many Unix shells.

Discovering the existence of `esyscmd` was the most time-consuming part of handling this.

## 6.2 When a comma is not a comma

Our most significant complication related to m4 was handling commas in arguments to macros. For example, we defined a command `_title` to typeset its argument as a title: it produces `\textsl{...}` for T<sub>E</sub>X, and `<i>...</i>` for HTML. The complication is when the argument contains a comma, as in `_title(The LATEX Companion, Second Edition)`.

m4 uses the comma to separate arguments; thus, the input `_title(A, B)` passes two arguments, ‘A’ and ‘B’ to `_title`. We don't want that. We want to pass one argument: ‘A, B’. The way this is done in m4 is to use quoting to make the argument into one string: `_title([[A, B]])`.

So, we added the `[[...]]` sequence where it was needed, and blithely continued on. Then one day we happened to notice that the ‘, Second Edition’ in the above title had disappeared from the output we were proofreading, and turned out to be missing in the generated T<sub>E</sub>X file. Being human, we had failed to add the m4 quoting and had instead written exactly the example above; m4 happily discarded the unused second argument that it saw.

Clearly we had to do an automated check. Besides titles, we had similar m4 commands to do



Figure 1: Our cover design.

footnotes, quotations, et al., which took lengthy arguments often containing commas. Since these commands could easily be split over input lines, we couldn't use `grep` or another line-oriented utility.

We implemented the check with the following code in m4 (and found a dozen or so places where arguments containing commas had not been quoted, so it was well worth it):

```
define([[_title]], [[dnl
  \ivtitle{$1}_iv_check_empty([[${2}]])]dnl
...
define([[_iv_check_empty]], [[dnl
  ifelse($1,, [[dnl no-op if arg is empty, else:
\misquote
errprint(__file__:__line__: misquoted $1.
)]]dnl end of ifelse
]])dnl end of _iv_check_empty
```

Explanation: each command such as `_title` passes its *second* argument to the helper `_iv_check_empty`, which then checks if its argument is empty and complains if not, using the GNU m4 `errprint` builtin command. (It took some time to get the m4 quoting right in the definition.) The `\misquote` command in the body is not defined in T<sub>E</sub>X, and is there just to make sure we can't run the book without fixing these problems.

All in all, m4 proved sufficiently flexible that we were glad to have chosen it.

## 7 Our cover

Our cover design (Figure 1) was inspired by the dust cover of the *Mathematical People* mentioned in section 1. That dust cover also included several photographs of interviewees, and the names of the interviewees were subtly listed on the cover. We did not match the subtlety, opting for a simple list on the back cover. We also mimicked that book's title.

The figure here is printed in black and white for *TUGboat*; for the color version that the book was

actually published with, see [tug.org/interviews/book/cover.pdf](http://tug.org/interviews/book/cover.pdf).

Dave did an early mockup of our cover using Illustrator, which he has used to create previous book covers. However, given the simplicity of our cover, we quickly decided to do the final version in T<sub>E</sub>X, and we copied the process of Yuri Robbers and Annemarie Skjold from issue 2007-1 of *The PracT<sub>E</sub>X Journal* for creating a cover using PSTricks ([tug.org/pracjourn/2007-1/robbers](http://tug.org/pracjourn/2007-1/robbers)).

Dave didn't want to take the time to figure out how to make the PSTricks gradient capability not crash on the computer he was using, so he still used Illustrator to create our cover's background gradient. He also used Illustrator to place a transparent version of the TUG logo for the front cover on the background gradient rather than trying to figure out how to do transparency with PSTricks. He used Photoshop for various manipulations of the cover photos.

Our PSTricks file for the cover is at [tug.org/interviews/book/cover.tex](http://tug.org/interviews/book/cover.tex). If you compare it with the PSTricks code in the Robbers-Skjold paper, you will see how closely we followed their template. The only addition we made was for our box of interviewee names on the back cover.

## 8 Development environment

We did much of the coordination of use of files in this project using Subversion. Although initially not sure that we needed a version control system, we were ultimately glad to have the notion of one "reference" master which we could both synchronize to and be sure we were working from the same point.

Dave works on Windows, and accessed the TUG server using programs Putty (<http://www.chiark.greenend.org.uk/~sgtatham/putty>) for basic command line work and WinSCP ([winscp.net](http://winscp.net)) for GUI work, among others. Karl works on GNU/Linux, and used the ubiquitous `ssh` and `scp` from OpenSSH ([openssh.org](http://openssh.org)) for server access.

## 9 Self-publishing

Self-publishing is becoming more widespread ([www.walden-family.com/self-publishing/](http://www.walden-family.com/self-publishing/)). Desktop publishing technology and Internet sales technology allows anyone who wants to make the effort to be his own publisher. There are two primary advantages: (1) the authors control the rights to the book and all decisions about how the book is published and distributed; (2) the authors don't have to find a "real" publisher and deal with publisher requirements about the design and content of the book. The individual or company which acquires and owns the ISBN number for a book is the publisher of record

for the book. The disadvantages of self-publishing are obvious: you don't have any of the guidance and services a publisher can provide and have to do everything yourself (or pay to have it done). We decided to self-publish.

Of the services usually provided by a publisher, we acquired the ISBN number (in TUG's name) and we felt confident about doing our own book design, typesetting and photo adjustments, and Barbara Beeton helped with editing. Between us we also had considerable experience producing, self-publishing, and distributing several books and a journal (*TUGboat*) via a variety of paths. Additionally, our plan was to submit the book to Lightning Source Inc. (LSI) for on-demand printing and access (via LSI's parent company Ingram Book Group, a giant book wholesaler and distributor) to large (e.g., Amazon) and small on-line (and any other) book stores.

Here's how self-publishing via LSI usually works. You submit a PDF file of the interior of the book and a PDF file of the cover formatted according to LSI's standards. This costs about \$120, and they send back a bound proof to review. You tell LSI the list price and a discount (typically 55% if you want good treatment by Amazon) for the book.

Thus, for instance, Dave's previous book had a list price of \$30 and a discount down to a wholesale price of \$13.50. When Amazon sells the book, it collects their price (typically discounted from list, say to \$25) plus shipping costs from the buyer. Amazon keeps the difference between \$25 and \$13.50 for itself and sends \$13.50 plus the money for shipping to LSI. LSI prints a copy of the book and ships it to the buyer, and deposits to Dave's bank account (a couple of months later) the difference between \$13.50 and its print cost (which is about \$6.25 for one copy of a 280-page 6 × 9 inch book).

Dave can also take a book order for \$30 plus shipping via his web site and PayPal, place an order to have the book printed and shipped to the customer by LSI, paying LSI (immediately) the print cost plus shipping cost, and end up keeping something like \$16 for himself. Given the price difference, and people's desire to deal with a known entity such as Amazon, Dave sells very few copies of his book via his web site and, thus, ordering and shipping the orders that come via his web site is not an undue burden.

While we want the interview book to be available via Amazon and other retail book stores, we also want to sell the book to members at a very modest price (one that does not allow a 55% discount between the list and wholesale prices), and we want to sell the book at essentially cost to interviewees. It's easy enough to use the LSI–Amazon path for non-

discounted sales. It is also easy enough to take orders via the TUG web site for members and interviewees, as long as the book is not so popular with TUG members that the volume of placing the individual orders on LSI's web site for drop shipping to members doesn't become a burden (although LSI supports automated electronic orders, it isn't available to low-volume publishers). We'll wait and see what our sales volume is to TUG members and what ordering options we can work out for LSI; making the book available via LSI does not preclude us from having the book printed by a second company that might lead to a simpler procedure for processing TUG member orders.

## 10 Going forward

We will continue to use HTML for the web site rather than the  $\LaTeX$ -based PDFs because HTML is by far the more effective format on the web. With all interviews in m4, it is now possible to use the Makefile and an expanded set of m4-to-html definitions ([tug.org/interviews/book/htmldefs.m4](http://tug.org/interviews/book/htmldefs.m4)) to more or less automatically regenerate the HTML-based web site using the interviews improved with Karl and Barbara's editing and the new endnotes from a few of the interviewees.

Even with this book published, we will continue to do interviews for the web site, but now using the various tools and processes we have developed as part of this book project. Maybe there will be a volume 2 of the book in another few years; if so, it will be easier next time.

We continue to appreciate the way the  $\TeX$  typesetting system can be smoothly integrated into a multi-step workflow process, as part of a typical distributed software development project involving multiple collaborators.

## Acknowledgments

We greatly appreciate the willingness of the interviewees to participate in the TUG interview series and to share their stories with the  $\TeX$  community.

Barbara Beeton volunteered to edit every interview, already a great gift to our interviews project. And how often does one have a first rate editor who also knows typesetting with  $\TeX$ ! Our appreciation to Barbara is unbounded. Thank you, bb.

Steve Peter was a kind source of answers to many miscellaneous questions.

Members of the TUG board, to whom we first announce each new interview we post on the web site, often catch typos in interviews.

Gianluca Pignalberi and David Crossland shared interviewing duties in two different cases.

The Computer Science Department at the University of Aarhus in Denmark supports the main  $\TeX$  Users Group server, where we post the interviews online and where we developed the book. Thanks to Kaja Christiansen, Michael Glad, and everyone there.