Multi-Use Documents: The Role of the Publisher

Kaveh Bazargan
Focal Image Ltd.
2 St John's Place
St John's Square
London EC1M 4NP
`kaveh@focal.demon.co.uk`

Abstract

Publishers now expect a variety of electronic files from the typesetter, both for publishing, and for archiving. And yet, the publishing industry, by and large, still follows the traditional "manuscript/edit/typeset/proofread/print" approach. The process is still essentially paper-based. The typesetting is also geared primarily towards paper output.

I suggest that we are in need of radical changes to these procedures so that files can be used to produce output on any medium, including paper, and I believe that these changes will benefit all concerned — authors, publishers, and typesetters. And clearly TEX is an ideal medium to hold the definitive text in modern typesetting. Not only can it be directly edited by the author, but it can produce every type of output required directly. These include paper, PDF, SGML, HTML, XML, etc.

Recent changes in typesetting author manuscripts

Some ten years ago, the process of typesetting material for publishers did not, in general, involve any form of electronic files, whether for typesetting or archiving. The author submitted a paper manuscript which was copy edited and sent to the typesetter for keyboarding, conventional proofreading, and setting to bromide. The procedure had evolved over decades, and if each party performed their roles and knew their responsibilities, the process worked well.

During recent years, the well-defined roles of the three parties — the author, the publisher, and the typesetter — have been blurred, due to electronic files, including those *submitted* to the publisher from the author and those *requested* by the publisher from the typesetter. When we look at current procedures employed by publishers, we see that in the main, the traditional manuscript-based approach is still taken, with the electronic files being treated as an afterthought, once the paper camera-ready copy (CRC) has been completed.

I would like to examine the use of electronic files, and propose some changes in the traditional procedure which should benefit all three parties.

Why use electronic files from the author? For the purposes of this discussion, I shall confine myself to TEX and LATEX files submitted by the author, although the principle applies to other file types.

There are two reasons why files submitted by authors should be used in typesetting a manuscript:

- By using the author's original code, typographic errors can be minimized. TEX documents are often complex mathematical or technical ones, and proofreading them is a difficult task. It therefore makes sense to use the author input where possible.

- If the author has used LATEX in a structured manner, then there may be a significant labour and therefore cost saving for the publisher.

Electronic files required from the typesetter. When publishers first requested electronic files from typesetters, it was limited to PostScript files. These were used firstly in order that the printer could produce high-resolution CRC to print from and, secondly, for archival purposes. Soon afterwards, PDF files were requested, having the advantage of smaller file size and screen viewability. After discovering the joys of hypertext links in PDF files, some publishers requested that these be included for citations, figures, tables, and sometimes to external URLs. Next came HTML and SGML, either for the full text or for abstracts and/or references. No doubt next in line will be XML, MathML, and who knows what will come after that. Gradually, these requests have increased the workload of the typesetter, often without any increase in prices charged.

Kaveh Bazargan

Using LaTeX to produce electronic files. Fortunately, we find that we can use LaTeX itself, in conjuction with many programs which are in the public domain, to generate all the electronic files required. It is the open code of TeX and that of the auxiliary programs written by third parties that allow us to use LaTeX in this way. The electronic files that can be produced in this way include full SGML files. An important advantage is that there is only one source code and the chances of differences between different electronic versions are minimized. Most of the tools needed for producing the electronic files are, in fact, in the public domain, with the source codes available. This means that they can be customized to produce specific electronic output, for example SGML output for a particular DTD.

The publisher's role

Here are what I think the publishers should be doing to help deal with electronic files better.

Take a more active role in encouraging good LaTeX submission. In order that files prepared by the author can be used easily for multiple outputs, it is essential that these files are coded with document structure in mind and the coding is completely logical, with little or no visual encoding. LaTeX, used correctly, is a very good authoring environment for this type of coding.

It is my feeling that the acceptance of TeX files by publishers has been under the pressure of authors, rather than being initiated by the publisher. With a few notable exceptions, publishers have simply passed on any TeX or LaTeX files received from authors onto typesetters, in case they can be used. By supplying the author with an "author kit" (see below) authors will gain confidence in the publisher and will be encouraged to submit LaTeX manuscripts according to the publisher's requirements.

Recognize the extra care needed when multiple outputs are required. More and more publications are available electronically, usually through the Internet. These include HTML, and PDF files. I believe that publishers should consider all forms of electronic delivery at the outset, and treat the printed CRC as just one of those possible outputs. They should take into consideration the extra care that should be taken in the preparation of manuscripts and their associated files. This means that they should be in contact with the author from the early stages, making sure that clean, structured files are submitted. If such files are not available, then the publisher should be prepared to compensate the typesetter for the extra work incurred.

They could also insist that all outputs, whether electronic or paper, should emanate from the same source code. This will guarantee uniformity in content of the different outputs and minimize delays for last-minute changes.

Distribute an author kit By being provided with such tools, the author is encouraged to submit a well-structured LaTeX document. Here is what the author kit could contain:

- the LaTeX manual [1]: this is a small investment which I believe will encourage the user to follow the standard) — it would only apply to book authors
- an authors' guide, based on the same user friendly approach as Lamport's manual
- a generic class file for authors only (see below)
- a fully working example file with accompanying hard copy

Produce an 'Editors' Guide to Electronic Submissions' Book manuscripts are normally sent to freelance copy editors before typesetting. The copy editors are generally unaware that there are files accompanying the text, let alone that the files may be TeX or LaTeX. It would be useful to produce a short non-technical guide for copy editors, in order to make the process of copy editing smoother and to reduce the number of marks made on paper. Points that should be addressed in this guide would be the following:

- *Automatic cross referencing*: It is useful for a copy editor to know if cross references for citations, equations, etc. are generated automatically. If an equation is deleted, for example, they can simply ask for the rest to be renumbered appropriately, rather than marking each occurance, which is the usual practice. They should also be warned of the reasons for mysterious double question marks appearing in a manuscript.
- *Table of contents and Index*: In LaTeX documents these are usually automatically generated. Therefore copy editing them is invariably a waste of time both for the editor and for the computer operator. In particular, it is very common for page numbers in index entries to be edited. For the typesetter, this is extremely laborious work to carry out and to check.
- *Running heads*: These are also automatically generated and should not generally be marked by the copy editor. By understanding the overall mechanism of running heads, a few simple instructions should suffice for any changes.

- *Making global changes*: The copy editor should be encouraged to make as many global marks as possible, rather than marking every occurance of an error.

The class/style files

The current situation. Let us take the case of book production. At present, a typical scenario is that the publisher asks a TeX consultant to write a class file (let us assume it is only LaTeX 2_ε we are dealing with). The consultant will be supplied with an example book and/or type specifications, from which to produce a LaTeX class file. Normally instructions and an example file are also included. This collection of materials is then distributed to prospective authors to apply to their manuscripts. When the manuscript is received by the publisher, it is sent out for copy editing, and usually goes to a TeX typesetter for production of final CRC and any subsequent electronic files.

The important thing here is that there is only one class file. It is used by the author to produce the manuscript and by the typesetter to correct and paginate the book. I would like to argue that this approach should be re-examined and that separating the author's and typesetter's class files might be a better route to take. Let us look at the requirements of each party in turn.

Requirements of the author. The author's task is primarily to concentrate on the contents of the book or article being written, without much regard for the final look, which is normally decided by the publisher and is the responsibility of the typesetter. Here are what I see as the main attributes of the class file distributed to the author:

- *Standard input syntax*: The LaTeX `book` class is well known, and easy to use. It is a great advantage to present an author with a class file which has an input syntax as close to `book.cls` as possible. In fact, the class file and any instruction material should encourage the author to enter standard LaTeX code.
- *Easy installation*: TeX is available on most computer systems, including some old machines with limited memory and power. It is a good idea to have a class file that does not need a lot of memory, computer power, or unusual fonts. In particular, it is safest to limit the font requirements to the standard Computer Modern fonts which are available on every TeX installation.
- *Avoid unusual elements*: In order to make it easy for the class file to be used with any system, unusual elements such as graphics should

be avoided and be reserved for the typesetting stage.

- *Forgiving class file*: So that the author can concentrate on the contents of the work, the standard TeX parameters for such items as line and page breaking penalties should be relaxed so that overfull boxes are kept to a minimum.

Requirements of the typesetter. Let us now look at the requirements of a class file used by the typesetter. In general, the typesetter should have a more in-depth knowledge of LaTeX than the author. He/she can therefore deal with much more complex class files, with the following possible attributes:[1]

- *Unusual fonts*: The TeX typesetter will have access to numerous commercial fonts. The class file can therefore use any standard font for the body text and a choice of several fonts for mathematical text.
- *Graphical embellishments*: I believe that typesetters should strive to get away from the classic "TeX look". There are numerous ways in which TeX and LaTeX documents can be embellished with graphical devices, from simple rules to full-colour tints. A TeX typesetter should have all these tools available to improve a book's appearance (in association with a book designer, of course).

Looking at the above requirements for the class files for the author and the typesetter, it is clear that there is a conflict. In the case of the author, the file should avoid non-standard fonts, difficult constructions, graphic elements, etc. On the other hand, the class file used by the typesetter can be as complex as necessary to get the desired effects in the final typesetting. At the moment, any class file writer has to strike a balance between these two conflicting requirements. Graphic elements, if used, are kept simple, so that authors do not have trouble with them. Of course, this limits the complexity of the final typeset book.

Using two separate class files. Looking at the above requirements, it is my conclusion that the author should be supplied with a generic class file, designed with the author requirements in mind. A publisher need not distribute more than one or two of these generic files, making support, debugging, and maintenance easier. The class file used in the

---

[1] Of course this may not apply to all cases. Some authors are very adept at handling class files, and some books have such an unusual layout that the final class file must be used at all times.

Kaveh Bazargan

final setting of the book need not concern the publisher at all but should be the responsibility of the typesetter.

Re-use of typeset files by the author

A unique advantage of using LaTeX to set books is that once the typesetting has finished, the files can be passed back to the author to work on another edition. This cannot be done when other systems are used for typesetting, as no other typesetting system can generate clean, structured LaTeX.

Sequence of events in setting a book with LaTeX.

Here is what I see as a possible sequence of events, regarding the LaTeX files used in setting a book.

1. Author signs contract with publisher.
2. Publisher sends author the LaTeX author kit.
   The author is encouraged to contact publisher for assistance at this or any subsequent point; typesetter could be called on to support the author in the interests of extracting clean files from the author.
3. Author sends sample chapter to publisher, typeset using generic class file.
4. Typesetter evaluates and reports on sample, with suggestions to author.
5. Author submits manuscript, accompanied by full set of files.
   Note: A possible extra stage here is that the typesetter reformats the files in the final style before handing over for editing, rather than the generic style file being used, exactly as submitted by the author; there are pros and cons to this stage.
6. Copy editor edits manuscript, having read the publisher's 'Editors' Guide to Electronic Submissions'.
7. Author reviews editing (optional, for books only).
8. Manuscript and files go to typesetter.
9. Typesetter implements editor's corrections, converts to final style, and sends copies to publisher.
10. Typesetter receives corrected proofs for CRC, produces CRC and any specific electronic files, and sends these to publisher and printer.
11. Typesetter strips out pagination codes used in the author files to make the final pages, and returns these files to author (via publisher).
12. Author uses generic style file to carry on working on other related material, such as the next edition of the book.
13. Go to step 5 to repeat cycle.

References

[1] Lamport, Leslie. *LaTeX: A document preparation system*, 2nd ed. Addison Wesley Longman, 1994.

The Journey of the TeXxies

"A cold coding we had of it,
Just the worst package from TeXLive
For a document, and such a long document:
The macros long and the braces many,
The very pits of TeX."
And the tables hard, badly-aligned, refractory,
Misplaced `omit` in every `\multicolumn`.
There were times we regretted
The dependence on longtable, the use of pdfmark,
And the hyperref macros breaking our `\cite` s.
Then the authors cursing and grumbling
And using Macintoshes, and wanting their
    Framemaker and Word,
And the editor crashing, and the graphics corrupted,
And the `setup.exe` hostile and CTAN unfriendly
And the usergroups dirty and charging high prices:
A hard time we had of it.
At the end we preferred to work in XML,
Validating as we went,
With the voices singing in our ears, saying
That this was all folly.
Then at `</tei.2>` we came to the end
    of the process,
valid, with all elements ended, all IDREFs satisfied,
With an XSL stylesheet, and even a version for IE5
And three HTML versions already on the Web.
And an old white horse galloped away in the meadow.
    —stolen by Sebastian Rahtz

(untitled)

I thought I would never see
poetry, written by Chimpanzee
But given TeX and infinite time
I'll steal it and call it mine
    —Donald Arseneau