**BIJLAGE S**

# TUGboat production: TEX, LATEX, and paste-up

**Barbara Beeton**

American Mathematical Society
Editor, TUGboat

## Introduction

*TUGboat* has now completed more than ten years of publication. Starting with TEX78 and an electrostatic printer and progressing through increasingly versatile software and hardware, the authors have kept us challenged, both with the content that the reader sees and the little tricks that happen "under the covers". This talk will be a survey of some of the milestones of *TUGboat* production, our editorial philosophy, what we've learned about what TEX can and cannot do, and some advice to authors and production editors of other publications.

## History

TUG and *TUGboat* came into being together. An inaugural meeting of the TEX Users Group was held at Stanford University in February 1980; it was attended by about 50 people. The reaction to this new tool—TEX—was immediate and enthusiastic. One of the first projects proposed for TUG was the publication of an occasional newsletter, and it was immediately decided to call it *TUGboat*.

Bob Welland, a mathematician at Northwestern University, volunteered to handle the editorial responsibilities. Production would be done at the American Mathematical Society (which, together with the Stanford TEX Project, had arranged the first TUG meeting); this task was given to me, as I had more experience with TEX than anyone else at the AMS, although I had never been responsible for either editing or journal production.

### The AMS before TEX / TEX before the AMS

A little "prehistory" might be useful here. The AMS had been using several proprietary photocomposition systems to prepare an increasing proportion of its research journals since about 1975, and for such "administrative publications" as its membership list and publications catalog for even longer. (The first typeset membership list appeared in 1971.)

Beginning in the early 1960s, the AMS had begun to investigate the possibilities for computerized composition of mathematical text. Most early computer-based typesetting systems were designed for newspaper production; many of them had quite strong pagination capabilities, but minimal ability to handle anything but ordinary text. Several systems appeared in the early 1970s that were able to deal with mathematical notation as well as text; however, they either required dedicated computer hardware or were unable to handle pagination, or both. In addition, all the typesetting software examined was proprietary, usually expensive (*troff* cost almost nothing for a university, but nonacademic users had to pay quite a high fee), and frequently had an obscure user interface.

The AMS was also looking at methods for incorporating large volumes of bibliographic material into a forerunner of today's database collections. For such a use, it was clear that logical, as opposed to typographic, tagging was the only sensible method for organizing these data. Even here the software must deal with mathematical notation—titles and abstracts of mathematical literature contain mathematical expressions, and to paraphrase or eliminate this notation could render much of the data meaningless.

In an effort to be part of the solution, representatives of the AMS (including me) participated for several years during the early 1970s in the GenCode project of the Graphic Computer Communications Association. GenCode was the forerunner of what later emerged as the Standard Generalized Markup Language—SGML.

At the annual meeting of the AMS in January 1979, the invited Gibbs Lecturer was Donald Knuth. The topic he chose to speak on was mathematics and computer science in the service of mathematical typography, and the new system he called TEX. The chairman of the AMS Board of Trustees, Richard Palais, was immediately attracted to the possibilities of the system that Knuth described. In addition to its ability to typeset mathematics and produce fairly sophisticated page layouts, this system was public, so it could be used by ordinary mathematicians who had access to computers at their universities.

The wheels were set in motion for AMS to become one of the first nonacademic users of TEX. This decision resulted in (among other things)

- acquisition of a new computer;
- support of a group to spend the month of July 1979 at Stanford for the purposes of
  - learning TEX,
  - bringing TEX back to AMS and installing it at AMS headquarters,
  - implementing TEX macros to produce two particular AMS publications: the Society's membership list, and the issue indexes of *Mathematical Reviews*;
- the development of AMS-TEX, a macro package intended to simplify input of complex mathematical expressions;
- the commitment to use TEX for all AMS publications prepared in-house.

## A very short history of TUG vis à vis the AMS

Within a year after TEX came to the attention of the AMS, it was realized that widespread support of this public-domain software would best be gained through a strong community effort. This in turn led to the formation of TUG. TUG was initially funded by the AMS and supported by AMS personnel, but after several years it had grown sufficiently strong to acquire its own personnel and incorporate as an independent society.

There has not been any formal connection between TUG and AMS since 1984. TUG still purchases some services from the AMS, such as the typesetting of *TUGboat*, and AMS personnel continue to participate in TUG meetings and as members of TUG's Board of Directors. However, the two organizations are now entirely separate.

## The first issue of *TUGboat*

Articles for the first issue were written mostly by people associated with the AMS, members of the Stanford TEX Project team, and their associates.

Some very primitive macros, based on `basic.tex` (the prototype of `plain.tex`), were developed to handle the formatting of a few obvious textual elements—title, section, etc. These macros were made into a style file that was used to prepare copy for most of the authors with AMS connections. A look at one of the article files created for the first issue shows a philosophy that hasn't really changed even today, namely the use of tags that identify objects, not coding to specify what they should look like.

```
\title Publishing \&\ \TEX\cr
\\Ellen E. Swanson\cr\end
 ...
\parhead The Manuscript.\end
 ...
\parsub Copy editing.\end
 ...
```

Explicit typographic coding was used in only three places in this article to obtain a desired format: in one itemized list, in the bibliography heading, and to obtain the desired vertical skip before the signature block at the end. This is typical of the articles prepared or processed at AMS.

Mike Spivak submitted an article describing his new macro package, AMS-TEX, which he was creating at the request of the AMS. AMS-TEX has its own "preprint" style, and Spivak's article was formatted with an adaptation of that style. Like the prototype macros for *TUGboat*, the preprint style uses mostly logical tagging, but the article reverted to raw TEX code for explication of AMS-TEX's input notation.

Processing at the AMS was done with TEX78, which was essentially Knuth's own version of the program; this was written in the SAIL programming language (which runs only on DECsystems 10 and 20), and converted by David Fuchs from the WAITS operating system on Stanford's DEC-10 to run on a DEC-20 under the TOPS-20 operating system. Output was prepared at 130% on a Varian 9211 (a 200dpi electrostatic printer that used liquid toner); this was reduced photographically when making the printing plates to improve the quality.

One figure, illustrating the position of the letter "q" in its font box, could not be generated with TEX (even now, the largest Computer Modern font available at the AMS isn't large enough), and was drawn onto the TEX-generated box in India ink from an enlarged photocopy. The cover art and the name "TUGboat" on the title page were likewise pasted up from non-TEX copy.

A letter to the TUG Chairman, Richard Palais, was reproduced from typewritten copy as received. (Nowadays, if we receive a letter only on paper, we convert it to a file and process it as if it arrived electronically.)

A long article describing an indexing facility for TEX was supplied as camera copy by the authors and included as an appendix to the first issue. The camera copy was prepared on a Versatec printer (another 200dpi electrostatic printer), and uses what appear to be Times Roman fonts rather than Computer Modern. I find this of particular historical interest because it shows that the use of non-METAFONT fonts with TEX is as old as TEX itself.

The *TUGboat* editor, Bob Welland, worked from paper copy prepared by the authors, or by the production staff at AMS (me). Nearly all communications were by paper mail or phone. Some files were provided on magnetic tape (diskettes did not become a reasonable alternative until much later). I was fortunate to have been "adopted" by the Stanford TEX Project to the extent of having guest accounts on the SAIL and Score computers, and thus some limited communication was possible with the Stanford TEX community and other authors who had Arpanet accounts. (I gained access to these guest accounts by long distance phone calls and Kermit. This was much more complicated than using the AMS Internet connection that is now available. Even so, it was definitely the method of choice for obtaining electronic files when that was possible for authors.)

Once all the editorial changes had been made and camera copy assembled, including pages for a separate mailing list, the copy was delivered to the printer and duly appeared in print in October 1980, eight months after the first meeting.

## Progress

The next few issues grew in size as TUG itself grew, and the affiliations of authors and the equipment available to them became more varied. *TUGboat* 2, no. (1) contained a self-referential article:

`\title` **HOW TO PREPARE A FILE**`\cr`
**FOR PUBLICATION IN TUGboat**`\cr`
`\\`Barbara Beeton`\cr`
American Mathematical Society`\cr\end`

This showed the tagging scheme used for the (somewhat improved) *TUGboat* macros, and encouraged authors to create their articles "on a computer file and submit [them] on magnetic tape." (We didn't yet offer to provide the macros to authors for their own use.)

Before *TUGboat* 2, no. (1) was sent to the printer, a new typesetter, an Alphatype CRS (with the phenomenal resolution of 5333dpi) had been installed at the AMS. It was not yet ready for extensive production, but one sample page was included in the issue, along with sample output of the same page from a variety of other (dot matrix, electrostatic, and laser) printers.

All the regular articles in *TUGboat* 2, no. (2) prepared at the AMS were output on the Alphatype. This included one article by Brendan McKay that demonstrated how very small dots (actually tiny `\vrules`) could be used in conjunction with ordinary rules and symbols from various fonts to create "pictures". The strain on memory capacity was severe (as it is now with PICTEX), but watching the output emerge from the film processor was well worth the effort.

By the end of 1982 I had become quite dissatisfied by the frequent necessity of either shrinking the type size of the contents to fit on the back cover, or continuing the listing on an inside page. *TUGboat* has always been subdivided into major sections, providing a logical structure for the contents, and in the early issues, the section headings appeared in the tables of contents above the listings of articles in those sections. Breaking the contents into two columns, with section headings to the left (right aligned) and article listings to the right (left aligned) eliminated enough extra lines to reduce the list to a single page. Unlike a book or report, *TUGboat* doesn't require numbered chapters or sections, so moving the page numbers to the left of the article titles didn't cause any visual conflicts; it also got rid of the dot leaders, which I wasn't particularly fond of either. The "new" contents style first appeared for *TUGboat* 4, no. (1); the final adjustment, moving the title and issue information from the top center so that it is left aligned with the article title listing, was made for *TUGboat* 5, no. (2).

## Changing of the guard

By the end of 1982, Bob Welland found that he was unable to continue as editor, and effective with *TUGboat* 4, no. (2), the title as well as the editorial responsibilities became mine. *TUGboat* 4, no. (2) incorporated two innovations in honor of this event. The first (which has not been repeated) was the accidental omission of the name *TUGboat* from the title page of the issue. The second was to number pages consecutively within a volume rather than starting over with each issue; not only did this simplify references to items from *TUGboat*, but it also eliminated a source of confusion when back issues are reprinted in full volumes.

The old SAIL implementation of TEX78 continued to be upgraded periodically at AMS to fix bugs. Since that was working well in production, we never attempted to install TEX80, the first Pascal implementation, and in fact, TEX78 continued to be used for AMS production until late 1986. This is not as backward as it may sound, for two related reasons. The first is that TEX78 was by then a stable system, while TEX80 was a prototype, with frequently changing syntax, and was almost certain to be replaced, as indeed it was.

The second reason for not upgrading more quickly was built-in delays in AMS journal publication procedures. AMS policy requires that authors be given a chance to review their papers after typesetting, and sometimes authors take many months to return the copy for publication; two years used to be the cutoff, after which a paper would be assigned to a journal issue for publication without approval. A typesetting program with unstable syntax is not suitable for use in such an environment.

TEX82 owes its name to the date when Knuth began the reimplementation of TEX in `WEB`, a process that was brought to fruition with the publication of the five-volume *Computers & Typesetting* series in May 1986.

The first use of TEX82 in *TUGboat* was in volume 5, no. 1, in an article by Knuth on TEX incunabula.[1] The proportion of TEX78 to TEX82 items dwindled for several issues, and the last article that required TEX78 appeared in *TUGboat* 6, no. (2) (July 1985, still nearly a year before the debut of *C&T*).

The gradual approach was taken to permit time for careful reimplementation of the *TUGboat* macros, not simply a translation from old to new syntax. The two are

[1] *Incunabula*, from the Latin for "in the cradle", designates books printed before a.d. 1501; hence works from the early period of a technology.

quite different, after all—the final list of differences between TEX78 and TEX82 takes 26 *TUGboat*-sized pages plus a three-page, three-column index.

## A few more milestones

At the same time, the entire appearance of *TUGboat* was re-examined, and suggestions solicited from anyone who turned up who seemed to have some design competence. Richard Southall's pre-TUG meeting short course in August 1984 was one source of ideas. (*TUGboat* had been offered, quite seriously, as a object for criticism, when that course was first being arranged.) The style of Southall's published lecture notes in *TUGboat* 5, no. (2), which is quite different from everything else in *TUGboat*, was intended to demonstrate the principles that he had expounded. This endeavor was relatively successful, as judged by Southall, failing principally in having centered text rather than asymmetrical margins. The particular style used for this article was not adopted in full, but the ways of looking at how conceptual structures are embodied in graphical form have been reconsidered every time a format change has been contemplated. I recommend this article and the references it cites for anyone who wishes to learn more about the traditions of typography, and why traditionally made books look the way they do.

One particularly interesting item from a quite different point of view is the set of benchmarks from a review of technical word processors made by members of the Boston Computer Society's PC Technical Group; this appeared in *TUGboat* 6, no. (3). The full review, which appeared in the *Notices of the AMS*, examined fourteen products (including two implementations of TEX) that could be used on IBM PCs and compatibles to prepare technical material for distribution. Of the fourteen, only four (including both TEX implementations) were able to produce the nine distinct benchmark samples, and only two of those (one TEX implementation failed) were able to combine all the samples and run them at once without a failure of either pagination or memory space. Furthermore, TEX output was judged to be in a class by itself, not surprising for a system designed to produce output of typographic quality. On the other hand, nobody said that using TEX would necessarily be easy, and the benchmarks demonstrate that. The file that produced this article is still one of the most stressful that has ever been processed for *TUGboat*, and I have recently provided a copy to be used in benchmarking the current crop of PC implementations of TEX; the results will be appearing in an upcoming issue.

## The guest-edited issue

*TUGboat* 7, no. (1) was a first in two ways: It was designed by a professional designer, and it was the first issue to use the `cm` versions of the Computer Modern fonts produced by METAFONT84. The time spent on this issue by the guest editors (David Kellerman and Barry Smith) and by me was more than double that required for any previous issue. Part of this is undoubtedly due to the development of an entirely new set of macros, and part on the fact that creation of the `cm` fonts for the Alphatype required double METAFONT processing—only METAFONT79 could generate fonts in the internal format required by the Alphatype, so 5333dpi bitmaps output by METAFONT84 had to be reprocessed for input to METAFONT79, etc., etc. The fact remains that the new macros, though more elegant and capable of a wider variety of effects than the old ones, required more time and attention in production. Since the position of *TUGboat* editor is a volunteer post, anything that increases the time required to produce an issue is counterproductive.

A couple of other important problems surfaced during the production of this issue. The first is the inability of TEX to know the attachment point of a footnote when a multi-column page has been divided by `\vsplit`. The second is more philosophical: What does one do when the style decrees that each article start on a new page, and an article ends with just a few lines on a page by themselves?

One other design question arose with the issue: What is appropriate content for the back cover? My own preference is to put the table of contents there, so that I can quickly find an item that I'm looking for. This serves such a useful function that there's no chance that an advertiser is going to get far asking for his ad to go on the back cover, even for a substantially higher price. But for *TUGboat* 7, no. (1), the designer wanted to keep the back cover blank except for the issue identification. This is a dilemma to which there is no objective solution, based as it is so clearly on personal preference.

In the absence of solutions to these problems, the next issue returned to the old style, but with a few small changes (such as in the format of the device driver charts) based on some easily adapted features of the new macros.

## And still more milestones

In spite of the fact that LATEX is the premier example of logically structured input for TEX, the first use of LATEX in *TUGboat* production didn't happen until *TUGboat* 8, no. (3), with the article on halftones by Adrian Clark. Before this, any articles submitted in LATEX were simply modified to work with the regular *TUGboat* macros. The main impetus for not converting Clark's article was the inclusion of a full-page figure (this figure and others were pasted in from copy prepared on a laser printer; the AMS typesetter was not able to produce them, and besides they were intended as examples of a particular software/hardware combination). This was a simple matter of expedience; the plain-based macros did not have an automatic method for leaving a blank page, while the LATEX macros did. The author helped out by providing the prototype of a style option that would make LATEX's `article` style produce pages that looked like

*TUGboat*. This style option has since been refined and extended so that it is now nearly impossible to tell which articles have been prepared with LATEX, other than those using the `doc` option.

*TUGboat* has always been a showcase for new TEX-related software and it's one of the joys of being editor that I often learn about new goodies long before most other TEX users. One of these new goodies was PICTEX, which was introduced in *TUGboat* 9, no. (2). The article appeared one day in my mail pile on an unsolicited diskette. It arrived complete with all the auxiliary software and test files, and even already used the *TUGboat* macros. The only problem was, it wouldn't run within the memory allocation assigned for the local production implementation of TEX, even though that was boosted about to the limit available with a half-word address space. Some severe pruning of the *TUGboat* macros, several hours of phone time talking to the author, and explicit page breaks in a few strategic places permitted the article to run to completion. The report from `\tracingstats` showed that, at the critical page break, only 8 bytes of memory remained untouched. This article has joined the Boston Computer Society benchmark examples in the test suite for the PC benchmark mentioned earlier.

### A hint of the future

With Frank Mittelbach's discovery of TEX, or more specifically, of LATEX, the proportion of articles prepared with LATEX, and more particularly with the new LATEX styles—`doc`, `multicol`, the new font scheme, and so forth—has blossomed. In *TUGboat* 1, no. 1(2), the most recent regular issue, more than half the pages, and two thirds of the articles were prepared with LATEX. We expect this growth to continue, and are looking forward to the new generation.

## How an issue gets put together

Until 1988, I had attempted to produce as much as possible of the camera copy for an issue in a single TEX run. I felt that this was a demonstration of how TEX was really supposed to be used. It was also a good way of flushing out unintended interactions between one author's macros and the next, and enabled me to post editorial warnings to unwary users. And, most significantly, I didn't yet have a production assistant. As soon as one goes from a solo operation to a cooperative arrangement, procedures have to change.

I should point out that the job of *TUGboat* editor is volunteer. I have a full-time job in the Computer Services Division of the American Mathematical Society, and though that is closely involved with TEX and other composition matters, the production of *TUGboat* is not included in its duties. Some work on *TUGboat* can be done during my regular working hours (and is billed to TUG), but most is done in the evenings and on weekends. In 1988, TUG hired someone knowledgeable in TEX to be the *TUGboat* production assistant. Early in 1989, Ron Whitney, who was formerly the head of the Composition Department at AMS, and with whom I had already worked on several major projects, became the TUG TEXnician.

The division of labor between editor and production assistant isn't always clear. In practice, Ron now prepares the first draft on paper and delivers it to me to read and comment. But if I happen to be checking the mailbox that day, I may get fascinated with an article that has just been submitted and do the preliminaries myself so that I can see it sooner. But both of us read every submission, and pay especial care if it's an article introducing new macros or a new LATEX technique. (After a gap of several years, we've recruited an associate editor for the macro column who should make this part of our job much easier. Victor Eijkhout's byline will appear for the first time in *TUGboat* 1, no. 1(4).) Although the editor has the final say (and should take the blame when something goes wrong), in the present arrangement we both feel responsible for trying to make *TUGboat* a publication we can be proud of.

### New macros for TUGboat

During the summer of 1989, Ron and I discussed at length what kind of user interface we really wanted. Needless to say, it would identify logical elements rather than typographic ones. We tried to regularize the syntax, and, for the plain-based macros, adopted some ideas from LATEX, in particular the bracket notation for optional arguments. As LATEX already provides notation for most of the structures that are likely to occur in a *TUGboat* article, all we had to do there was make sure that the formal characteristics of the output conformed to the *TUGboat* style. (It was also necessary to subvert some of the LATEX `article` style conventions to permit multiple articles to be run together, but that affects only the *TUGboat* production staff, not authors.)

Then Ron took apart and reconstructed the plain-based macros to obtain the desired effects, and we wrote an article that not only instructs authors how to use the new macros, but was also the stress test to demonstrate that they worked. Instead of writing a separate set of instructions for the LATEX style option, we included a section in the article that describes the distinctive macros for *TUGboat* top matter, and refers authors to the LATEX manual for everything else.

We now expect authors to use the macros provided. We try to keep an up-to-date version installed at the major archives, and we will send copies on diskette to authors who have no network contact. It doesn't really matter which version an author uses—plain or LATEX; even though we can't mix them in one run, we have devised simple and reliable TEXniques for starting an article anywhere on a page, just by providing the information on where the previous article ended.

There are still some legitimate reasons for authors to provide camera-ready copy instead of (or in addition to) sending us the files. The most common reasons are

- requirement for special fonts (e.g., Japanese or pointing hands), although we can do and have done extensive font work when METAFONT sources were provided with enough lead time;
- the intention of showing off the quality of the copy produced by a particular output device;
- the inclusion of an illustration that is relevant to the article, but that wasn't prepared with TEX or requires hardware that isn't available to the production staff.

The joints that result from discontinuities between plain and LATEX articles are patched, non-TEX insertions are pasted in, and the final touches (like running heads) on copy provided in camera-ready form are added by hand, with the help of a razor blade and waxing machine. We don't believe that, in most cases, readers can tell the difference between items prepared in one way or another. But we do think it's important to identify the different TEXniques used, so we provide a **Production notes** column in every issue.

### Help for authors and readers

As one of our post-production chores, we try to send back to every author the actual file used for the article as published. (We also send a new copy of the style files if they've changed.) When editing the article files, we try to comment out the important bits that we've changed, rather than simply making changes, and to add comments where we think our intentions may not have been clear. Then the author can see what we consider good practice, and might learn from it.

We also think it's very important to point out to both author and reader any special production problems. I've already referred to one such problem, with PICTEX and limited memory.

Another problem that occurs more frequently than we'd like is the redefinition by some authors of either plain control sequences or even TEX primitives. Surprisingly, most primitives aren't protected against redefinition, and even experienced TEXers often don't think about the consequences of what might happen if they choose an "obvious" name for some macro, and it just happens to be the name of a macro that's intimately involved in, say, the output routine. (Try the definition `\def\vbox{\hbox}` and a short paragraph or two.) Most re-uses of macro names aren't quite so obvious, though, and users who pick up some new macros they've just read about without considering the possibility of name conflicts could be in for a big surprise if no warning has been given.

## What have we learned?

*TUGboat*, in order to be a useful publication, must be self-referential. That is, the content of the articles will often describe the techniques used to produce the articles.

### We have to keep ahead of the authors

*TUGboat* authors are highly inventive. For example, TEX wasn't designed to support pictures; that's what the `\special` command is for—it "enables you to make use of special equipment that might be available to you, e.g., for printing books in glorious TEXnicolor." But authors seem to ignore that small restriction, and figure out ways to prepare graphics within TEX, without `\specials`. Even Donald Knuth has ignored his own advice and developed a variety of fonts to be used for halftones (*TUGboat* 8, no. (2)). Nobody has yet submitted an article that requires color printing, but I guess we should expect it. (An article by Ken Yap, *TUGboat* 1, no. 1(2), does tell how to prepare color slides.) And I am quite sure that the day will come when such an article appears in the *TUGboat* mailbox that *doesn't* require `\special` processing but does ask for color.

The moral of this is that the editor had better be prepared for anything. It's probably true that a publication like *TUGboat* can't be edited successfully by someone who is less familiar with the use of TEX than the average author, and the authors are getting a lot more clever.

### We have to stay organized

Attention to procedures shouldn't be forgotten. We've learned that communication between members of the production staff is absolutely essential. It's no fun to find that the changes you just made to the grammar in an article have just been obliterated by equally necessary updates to the macros made by the other worker. And it's also a big shock to find that the macros that worked yesterday aren't recognized today, because a new version of the style file has just been installed. Even if it's a one-person operation, it's necessary to keep careful records, so you can identify exactly which of several versions is the one that's to go to press.

### Let's not forget our typographic heritage

I'd like to make one final point: TEX users, both authors and readers of *TUGboat*, may not be familiar with the traditions of fine typography. We have a tool that is capable of being used with great precision and craftsmanship, if only we know what models to follow. It is one of my goals to try to educate this audience, myself included, in the best and most appropriate ways to use this tool. The editors of the Seybold Report on Publishing Systems have stated the case well:[2]

[2] Seybold Report on Publishing Systems, Vol. 19, No. 22, August 20, 1990.

Over the years, in spite of the remarkable productivity gains that it has produced, the computer revolution has left its mark in a negative way: the craftsmanship that went into certain parts of composing type has been sacrificed. We're not suggesting that we go back to the old ways, but we are making a plea toward slowly raising the current standard through increasing awareness of the issues and opportunities.

**BIJLAGE T**

# SGML and TEX at Elsevier Science Publishers

**Jeroen Soutberg**
Elsevier Science Publishers
Amsterdam, The Netherlands

Augustus 31, 1990

## Contents



## Introduction

SGML and TEX are in use at Elsevier as production tools for an increasing variety of products. In the following we describe the prevailing trend in publication technology, and the roles SGML and TEX play in the developments at Elsevier.

## ESP

Elsevier Science Publishers is one of the larger scientific publishers in the world. We publish about 600 journals and annually about 600 books in several fields of science. The publications appear under two imprints: Elsevier and North-Holland.

### Elsevier

The Elsevier imprint is used for publications in biology/medicine, chemistry, agricultural and earth sciences. Well-known examples are:
- Biochimica & Biophysica Acta,
- Journal of Chromatography,
- Ecosystems of the World (book series).

### North-Holland

The North-Holland imprint is used for publications in physics, materials science, engineering & design and mathematics and linguistics. Examples are:
- Nuclear Physics (A, B),
- Surface Science.

## Manuscript routing

The publishing world is in transit from a state of established procedures and practices to a state of constant adjustment to the technical and procedural opportunities arising from technological developments. These changes have a profound effect on the routing of manuscripts in the publication process.

We give a simplified description of this transition by splitting it up into three stages:
- The traditional situation;
- The present – transitional – situation;
- The future: database publishing.

### Manuscript routing – traditional

Traditionally there have been two ways of producing a printed publication: (1) Preparing a typeset publication from the manuscript submitted by the author, and (2) printing a publication from camera-ready material prepared by the author.

These scenarios yielded quite different print qualities, but correspondingly different costs were involved. These differences made it relatively easy to choose a scenario for a project, depending on its specific requirements.

The traditional scenarios are shown schematically in figure 1.

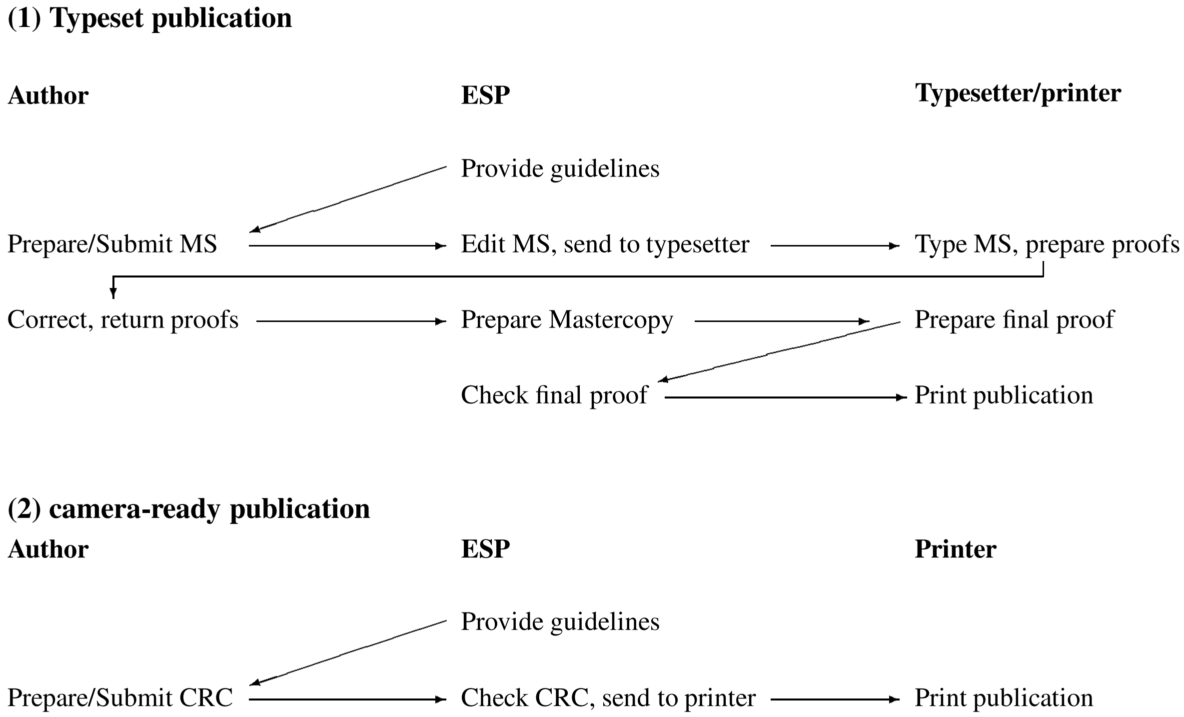


Figure 1: Traditional scenarios for printed publications

## Manuscript routing – present

The advent of electronic text processing (in typesetting systems as well as in 'word processors') and the availability of laser printers has upset the clear distinction between the traditional scenarios.

On the one hand, the CRC output an author may produce using a text processing system or a DTP package is much better than in the past, and it may even approach the quality of typeset material. On the other hand both authors of traditional manuscripts and authors of CRC material may want to submit their information in electronic form, in the first case for saving proof-reading time or accelerate publication, in the latter because they do not have a high-quality output device available. In both cases the publisher will have to process an electronic 'manuscript', possibly with media and/or code conversion. Also, the boundary between publisher and typesetter becomes less clear (who does the conversion, who edits the files, . . . ).

Another trend in publication is the preparation of secondary information from primary information and the re-use of primary information in new products. It is desirable for these purposes to have the primary information available in electronic form.

Figure 2 shows the schematics for typeset and CRC material with the conversion stage and re-use taken into account. In comparison with figure 1 the distinction between typeset and CRC material is much less clear; theoretically it is now even possible that a typeset and a CRC publication end up following the same routing. Also, these are not the only products anymore.

## Manuscript (information) routing – future

Extrapolation of the trends sketched above leads to a situation where information reaches the publisher in many different formats, and is processed into many different end products. The simple distinction between typeset and CRC material then loses its relevance.

It is not feasible to construct bridges ('translation programs') between all of these input and output formats. Apart from the number of programs required (number of input formats × number of output formats) the need to store information for future re-use would imply storage facilities for the same number of formats.

The ultimate solution to these problems is the concept of 'database publishing': the publisher has all incoming information converted into *one* intermediate format; the information is stored in a database in this format, and all end products are manufactured by making a selection from the database and converting this information into the appropriate format. In this way we have one standard storage format, and the number of translation programs needed reduces to the sum of the number of input and output formats. Figure 3 presents a schematic picture of this concept.

The publisher has now become an 'information broker'

**(1) Typeset publication**

| **Author** | **ESP** | **Typesetter/printer** |
|---|---|---|
| | Provide guidelines | |
| Prepare/Submit (electronic) MS → | → Conversion to 'final' products . . . | |
| Check proofs → | Send to printer → | Print publication |

**(2) camera-ready publication**

| **Author** | **ESP** | **Printer** |
|---|---|---|
| | Provide guidelines | |
| Prepare/Submit CRC + electronic MS → | Possible conversion/correction of EMS to 'final' products → | Print publication |

Figure 2: Transitional scenarios for printed publications

with all its valuable material stored in the database. Correspondingly, an author now submits information for the database.

The fact that many different outputs are to be generated from the intermediate format poses strict requirements for this format:

- It should be *presentation-independent* (e.g., the possibilities to identify a heading by its presentation are quite different for printed publications and on-line applications, so the database should contain the heading identified as such; not less, not more).
- It should be *consistent and clearly defined* (e.g., every identification should identify the same, well-defined piece of information).
- It should be *internationally accepted* (to make it possible to exchange information with others).
- It should be *implementable* (e.g., software and hardware available).

The requirements for the intermediate format affect the instructions to authors as well: now the material produced should be in a format which contains sufficient information to generate the intermediate format. This could mean, for instance, that references should be submitted in a special format which can be converted automatically for the database.

## Role of SGML – the intermediate format

Over the last few years SGML has emerged as the primary candidate for storing information in the manner described in the previous section. It goes a long way to meeting the requirements for the intermediate format introduced there. In a different order than above the requirements are met as follows:

- SGML is internationally accepted, an ISO standard.
- SGML is implemented in processing software like Softquad, Writer Station, et c.
- To a large extent, SGML is consistent and clearly defined; over the last year we have found some problems which we dealt with by creating preferred versions of ambiguous solutions for use within Elsevier.
- SGML is meant to be presentation independent. In the last few years a discussion has been going on within Elsevier about the question how far one should go in this. For example, the simple equation $s = v_0 t + \frac{1}{2}at^2$ cannot be presented in presentation-independent form unless the entire formula is spelled out. Even then it will be difficult to make the factor $\frac{1}{2}$ presentation-independent. The same problem arises in all instances where *for reasons of efficiency* a special presentation was introduced for identification purposes. Our present projects adhere to presentation independence down to the level of the *structure* of the information.

| **Author** | **ESP** | **'Typesetter'/printer/database/...** |
|---|---|---|
| | Provide guidelines | |
| Prepare/Submit electronic information → | Conversion of information to intermediate format ... → | Present information in many different formats |

Figure 3: Future scenario for information processing

## Role of TEX– Input format, output format

At Elsevier, TEX is used mainly for two rather different classes of applications:

- Where SGML is used as the intermediate format we use TEX as (one of the) output facilities. This permits us to generate, among others, high-quality paper output of proof material formatted in such a way that the information loaded into a database can be validated without looking at SGML codes.
- TEX is also used as the combined input–output format in projects where compuscripts from authors are processed in TEX all the way through. The conversion of TEX to other formats – which needs considerable manual correction – is thus avoided. Also, high-quality output can be generated locally.

## Projects

As examples of the different classes of applications mentioned in the previous section we describe two ESP projects representing these classes

### CAPCAS

The abbreviation, CAPCAS, stands for Computer-Aided Production of Current-Awareness Services. Although the system can do more or less what its name implies, its status has changed with developments. It is now intended to be a first step towards database publishing.

The scope of CAPCAS is at present limited to the storage of the Heads of journal articles in a database. (By Head we mean the bibliographical information and the abstract at the beginning of an article). Article tails and bodies will be dealt with later. As explained above, the information in the database is stored in SGML format.

**Input** is provided by our typesetters. This is at present the easiest input route, because a vast majority of articles are still being submitted on paper and the information becomes first available electronically after key-in by the typesetter. The typesetter prepares both the files for the typesetting equipment and the SGML files for CAPCAS from the key-in.

**Output** is generated for in-house and external applications: in-house for validation of information loaded into the database, contents lists and indexes, externally for secondary services.

- The validation of information loaded into the database concerns mainly the correct identification of fields in the Head, especially those which do not show up in the printed article (e.g., city and country are not typographically distinguished from the organization name in the affiliation). After loading the Head of an article into the database a so-called 'load report' is produced using TEX, whereby all the SGML tags generate unique layout features. As a result, the fields may be identified by simply looking at the layout instead of checking the positions of tags.
- Contents lists and indexes are generated from the database in SGML format. One of the present options is to produce output with the help of TEX, making use of the typical TEX 'programming' possibilities for handling information.
- As yet, no products are being delivered to external customers: we only just started loading the database. Also, before any products may be delivered the exact specification has to be established in consultation with the customer.

**Future** developments will be the incorporation of the other constituents of articles: tails (i.e. reference/note section), and (ultimately) the bodies (main text, including formulae, tables, figures).

We will start working on the tails fairly soon (probably in the coming year); the body, by sheer size and complexity, will not be tackled for some time to come.

### LATEX compuscript processing

Similarly to other publishers, there are several Elsevier journals where most manuscripts received are coded in TEX or LATEX. A project has been set up to create an article routing in which the electronic version of the manuscript is processed, keeping within TEX, and the output is produced with TEX as well; proofs are printed on a laser printer while the final output is produced in high resolution on the in-house typesetting machine. Special software makes it possible to generate output using 'Times' fonts instead of cm fonts.