

COLOPHON

The attached paper is another in a series of experiments to see how long it takes me to re-build electronic versions of my published early papers as properly re-typeset ‘PDF Normal’ rather than just as a bitmap scan.

The attached paper appeared in a special issue (Vol. 32, No. 6, pp. 482–493, December 1989) of *The Computer Journal* devoted to Electronic Publishing.

The original text of this paper was prepared in UNIX *troff* and the source file was sent to Cambridge University Press (CUP) who then re-typeset the paper. The published version of this paper is believed to be available as a bitmap scanned PDF, via the British Computer Society Web site. However, to see the paper in that form it appears that an individual or site licence will have to be taken out with the current publishers of *Computer Journal* (Oxford University Press).

The rebuilt version here represents the ‘final draft’ form of the paper before it was sent off to CUP. The pagination is different to the published version but the text and diagrams are identical. The diagrams were included using the UNIX pre-processors *tbl* and *psfig*. The time taken to rebuild the paper was about 1 hour.

© D.F. Brailsford and R.J. Beach 2005

‘Electronic Publishing’ — a journal and its production

David F. Brailsford

Department of Computer Science
University of Nottingham
NOTTINGHAM NG7 2RD
UK

Richard J. Beach

Xerox PARC
3333, Coyote Hill Road
Palo Alto
California 94034
USA

The use of the term ‘Electronic Publishing’ transcends any notions of ‘the paperless office’ and of a purely electronic transfer and dissemination of information over networks. It now encompasses all computer-assisted methods for the production of documents and includes the imaging of a document on paper as one of the options to be provided by an integrated processing scheme. Electronic publishing draws heavily on techniques from computer science and information technology but technical, legal, financial and organisational problems have to be overcome before it can replace traditional publication mechanisms. These problems are illustrated with reference to the publication arrangements for the journal Electronic Publishing — Origination, Dissemination and Design. The authors of this paper are the co-editors of this journal, which appears in traditional form and relies on a wide variety of support from ‘electronic’ technologies in the pre-publication phase.

1. Introduction

Over the past five years electronic publishing (EP) has changed from being a topic of minority interest into one of the fastest-growing areas of applied computer science. However, in describing it this way — as ‘applied computer science’ rather than ‘a computer application’ — we are claiming that electronic publishing relies heavily on the techniques of computer science itself. Those interested in a preliminary essay on this topic will find something in Reference [1]; but in this paper we want to widen the discussion, and to make clear that successful EP also makes considerable demands on what can best be termed ‘integrated information technology’.

To begin with we need a working definition of EP. At the risk of sounding too ambitious this definition has to encompass **all** computer-assisted methods for the production, manipulation and dissemination of information. This includes such specific topics as book and newspaper production, technical documentation, word processing, ‘Desktop Publishing’, document production from databases, hypertext, structured document editors, documents on CD-ROM, document previewing on workstations, digital typography, teletext, viewdata systems, electronic mail and bulletin boards. This list is a potent mix of multi-disciplinary studies, multi-media hardware and multitudinous headaches for anyone wishing to combine these themes into some seamless and integrated view of document manipulation. It would be pleasing if we could report that all the difficulties in EP melt away under the onslaught of well-tried algorithms from computer science with a judicious admixture of graphics arts expertise from designers and typographers. But the truth is that the abstractions needed to integrate all the topics in our list are only dimly apparent. So this paper will not be offering ‘expert systems’ for electronic publishing nor hypothetical solutions cloaked in pages of formal proof. All we shall attempt is a brief account of two previous experiments in electronic publishing, followed by a description of the techniques we ourselves

use to produce a specialist journal about EP.

A journal for EP

Having decided, in 1987, that EP was sufficiently active and important an area to merit a journal of its own, we enlisted the support of the Chichester (UK) office of John Wiley Ltd. as publishers for a new journal entitled *Electronic Publishing—Origination, Dissemination and Design* ('EP-odd' for short). The arrangement is that one of us (Brailsford) acts as UK/Europe editor and editor-in-chief and the other (Beach) as the US editor. Although the subject matter of *EP-odd* ranges over all the topics listed in the previous section we decided to stay with a traditional hard-copy published form, for reasons that we shall elaborate on later, but to use as many aids of modern information technology as we could in the pre-publication stages. By building up an archive of published papers as marked-up source texts, and by retaining the output form of the papers in the PostScript[®] language, it will be easy for us to move to full electronic dissemination at some later stage. To date we have produced a pilot issue (in January 1988) and four regular issues of the journal

It may come as a surprise that we intend to illustrate the problems of electronic publishing simply by relating various decisions about the forms in which papers for *EP-odd* can be accepted and by describing how we produce the journal. One would not expect engineers and scientists, for example, to illuminate their field of research by relating the problems of producing a learned journal in that same field! However, the last two sentences of our editorial in the pilot issue [2] summed up our bemusement at the self-referential world in which we found ourselves:

“This journal about electronic publishing reflects a unique situation: the scope of the journal encompasses the process for producing the journal. The message is the medium.”

Since that time there has been a continuing and spirited debate between editors and publishers as to how the journal should be produced, and in what form it should appear. Given that our authors, by definition, wanted to publish 'electronically' what exactly did this entail? Would there be expectations of instant publication via mail networks with no need for any hard-copy version? Worse still, if we did decide to have a hard-copy version, would it end up as a repository for the most complex papers, (involving half-tones, exotic fonts and so on) which were hard to transmit electronically? If this happened then the hard copy version of *EP-odd* would become the perfect embodiment of Russell's paradox ('*Electronic Publishing* publishes only those papers on electronic publishing that are not amenable to electronic publishing').

In the end some recent advances in two key technologies materially affected the choice of production methods used for *EP-odd*, and we start by devoting brief sections to each of them. We then go on to look at two experiments that have already taken place in producing 'electronic journals' and use these to explain the cautious compromises we have adopted. The present mechanisms for producing *EP-odd* are described and we explain why its canonical form is still that of a hard-copy conventional journal, despite the extensive use of modern 'electronic' mechanisms for acquiring, refereeing and cataloguing papers. Section 4 sets out the choice of text-processing software and the reasons for rejecting some other possible solutions for producing the journal. We also explain the need for imposing structure on documents and why it is that markup-based software, rather than the 'what you see is what you get' (WYSIWYG) sort, has found greater favour so far. Finally we review our experiences in producing the first four issues of the journal and survey the problems to be solved before *EP-odd* can become a fully electronic journal.

2. Two significant developments

In recent times one particular technical advance —the popularity of electronic mail— seemed at first to be pulling EP in the direction of paperless publishing. However, the availability of laser-printers, with output of near-typeset quality, has caused a swing in favour of computer-aided publishing on paper. In order to understand what has taken place we devote a short section to each of these developments.

2.1. Electronic Mail

Long-haul computer networks owe much to the pioneering work done on the US Department of Defense ARPANET project from 1969 onwards. The physical network now ranges from Hawaii to Sweden and uses a mixture of satellite links and high speed leased lines. Spurred on by the success of ARPANET a variety of private and public networks has grown up with particularly marked growth over the last five years. Within the academic community networks such as CSNET, JANET, BITNET and EARN are extremely popular [3] and some of them still use the software protocols developed for the ARPANET. Perhaps the largest network in the world is that formed among UNIX[®] users —the UUCP/USENET network— which links more than 10 000 computers and around a million users.

The networks vary considerably in the sophistication of the services they are prepared to offer. The most basic and frequently offered service is that of electronic mail (e-mail) which, if all the networks are functioning, can usually deliver an electronic letter to the other side of the world within a few hours.

For a journal such as *EP-odd* e-mail is invaluable for the exchange of information, on an informal basis, between editors, editorial board members, authors, referees and production staff. However, as a generalised means of electronic publication, it is far from satisfactory. Mailer software was never intended for the transmission of lengthy papers and some computers on the e-mail networks try to guard against this sort of abuse by limiting the size of ‘letter’ that they are prepared to pass on. Moreover any publisher using e-mail has to take active responsibility for posting material to subscribers, and this requires a carefully compiled and closely targeted list of recipients. Electronic junk mail is even more infuriating than the conventional sort and the posting of bulky academic papers to unwilling recipients would soon give any electronic journal a bad reputation.

Among the networks mentioned, USENET has been one of the pioneers in offering an *electronic news* service in addition to electronic mail. The news is divided into several hundred newsgroups and USENET users can selectively subscribe to those newsgroups which interest them. The news is copied from machine to machine on USENET using crude but fairly effective mechanisms for limiting duplicate copies. This level of network service enables a form of electronic publishing to take place and we take a closer look at this in section 3.2.

2.2. Laser-printers and PostScript

During the same period that computer networking and e-mail became commonplace there were also major advances in the quality of material that could be produced on laser-printers under computer control. The relatively low cost of laser-printers comes about because they share a common technology with the latest generation of photocopiers. To be more specific, it is relatively straightforward to convert a photocopier into a laser-printer by replacing the platen and the optical system with a single-board microprocessor that can interpret output from a host computer. If this output is in a suitable form to be interpreted by the software on the controller board then pages of text and graphics can be imaged on the photocopier drum. Much of the early work in this area, both in manufacturing laser-printers and in developing software to drive them, was carried out in the Xerox PARC laboratories and has been chronicled by Brian Reid [4]. It is significant,

too, that laser-printer technology is used in modern typesetting machines, which also form an image by raster scan of a laser beam. In the laser-printer case, the final output medium is paper, and the resolution is about 300 lines per inch, whereas a typesetter produces output on film, or on bromide photographic paper, at a resolution of 1000 lines per inch or better.

The software embedded inside the laser-printer determines to a high degree the sophistication of the images that can be formed. Simple software systems give rise to laser-printers with a choice of fixed-pitch typewriter-like fonts—in effect a substitute for the daisy-wheel printer. More complex interpreters implement some form of *page description language* and these allow for greater precision in placing characters on the page and for a wider range of typeface families and graphic effects. The most significant development in this area occurred in 1983 when two workers at Xerox PARC, Warnock and Geschke, left Xerox to form Adobe Systems Inc and to develop an interpreter for a page description language called PostScript [5],— so named because it uses reverse Polish notation. The low-level nature of PostScript makes it difficult to use directly and it is more usually encountered as the output language from a variety of typesetting software. There are built-in facilities for producing rasterised text characters, ‘on the fly’, from character outlines stored as Bézier splines, together with primitives for lines, circles, arcs, ellipses, splines, half-tones, grey-scales and so on. The fact that the PostScript language uses only the printable ASCII character set is a great help in tracing faults, but gives rise to some very large files of output.

PostScript has now become a *de facto* standard as a page description language and is a potent unifying force. It is largely independent of output device resolution, which enables material to be proofed on an 80 dots per inch workstation screen (running a ‘display’ form of PostScript), or on a laserprinter at 300 dots per inch, and then to be transferred, without fuss, to a 1200 dots per inch typesetter, provided that this also has a PostScript interpreter.

The wide availability of PostScript laser-printers has made possible an entirely new approach to the dissemination of documents. Indeed it was the combination of the Apple Macintosh machine, with ‘front-end’ page-layout software, linked to an Apple Laserwriter with an embedded PostScript interpreter, which gave birth to the era of ‘Desktop Publishing’ (DTP).

3. Two previous experiments with electronic journals

The arguments for a truly electronic journal, freed from the constraints of a hard-copy format, are very powerful. Given that communication costs can be held down it would be possible to distribute papers via the existing computer networks and to provide a large amount of value-added service to go along with them. Database and archiving systems could be developed for the journal and made visible to its subscribers. With international co-operation, these schemes could be extended to cover several publications and users could then browse through some centralised database to find papers of interest. Within any individual paper further browsing could take place in a non-linear manner and hypertext links could be inserted to aid the acquisition of information from the document base. In the limit it has been conjectured that the whole of the world’s information resources might become one vast hyper-document [6].

A more down-to-earth argument in favour of electronic journals is that the production and distribution costs of small specialised journals are ever increasing. University and public libraries find it difficult to afford subscriptions to the large number of titles that are now available. The first experiments with ‘truly electronic’ journals, which took place several years ago [7, 8], focus on interactive teleconferencing and discussions via electronic mail. It has to be remembered that these experiments took place in an era where there was much talk of the ‘paperless office’ and where laser-printers and typesetters were sufficiently expensive as to be quite out of reach for the average computer user. In more recent investigations there is a growing acceptance of good-quality hard-copy output as being part of the framework of EP.

We now describe two experimental EP systems, one centred in the UK and one in the US, which have served as benchmarks for some of the procedures we employ in the production of *EP-odd*.

3.1. The BLEND project

This project was funded by the British Library and ran from 1980 to 1984. A community of some fifty workers in the Computer Human Factors field agreed to submit articles to an electronic journal which would be coordinated using Infomedia Corporation's 'Notepad' software running on a DEC 2060 computer [9, 10]. The scheme provided an informal message exchange, pre-publication of drafts of articles, a newsletter and a general purpose electronic mail service. Formal publication of the journal, on paper, took place from time to time and, during the period of the experiment, the subscribers contributed 88 articles ranging from short notes to formal reports.

The investigators on this project suffered all the tribulations faced by those who pursue a good idea ahead of its time. They found, early on, that readers wanted paper copies of articles that could be studied in the train, or taken home. At the time of the BLEND project e-mail was still in its experimental stages and the only output devices generally available were dot-matrix or daisy-wheel printers, which were incapable of doing justice to any graphics, half-tones or illustrations that might have been part of the paper. PostScript laser-printers, which could have addressed all these problems, only began to appear as the project ended.

In a survey of experience gained from the BLEND project [11] Dodd concludes that a good compromise is a 'hybrid journal' which receives and processes papers electronically but retains a traditional ink-on-paper format for the definitive published copy.

3.2. The USENET Cookbook

Since December 1985 Brian Reid has published an incremental book of recipes, over the USENET network [12]: this experiment in EP is known as 'The USENET Cookbook'. A distribution of new recipes takes place every Thursday and, for those subscribers who do not have easy access to network news, there is a limited capability for receiving the recipes by direct electronic mail. Although the subject matter of The USENET Cookbook is less formal than that in the BLEND project the number of subscribers and participants (estimated at 13 000) is much greater.

In many ways recipes are the ideal choice for this sort of system. They are short items, capable of being produced in a simple standard format, with no need for complex graphics or half-tones. Subscribers receive the recipes electronically but may choose to browse through the latest ones on a terminal and to print out those that are of special interest. The hard copy of the recipes might be filed away in some sort of loose-leaf folder with old recipes being discarded from time to time. Two of the overriding problems with electronic dissemination — enforcement of copyright and extracting money for the delivered information — were neatly circumvented. Recipes do not have copyright protection in the US and Reid's employers were prepared to support the costs of The USENET Cookbook as an experiment in electronic communications.

Although The USENET Cookbook started only 18 months after the BLEND project ended, the technology of computer networks and of laser-printers had progressed enormously in that time. Furthermore, Reid benefitted greatly from choosing USENET as his distribution vehicle because most of the machines on that network run UNIX. Consequently the UNIX typesetting package called *troff* would probably be available, if only because it was bundled in with early UNIX distributions. (An extra-cost version, called *ditroff*, is recommended if a wide range of output devices is to be driven and a back-end driver package, such as Adobe's TranScript[®], is needed to convert the *ditroff* output into PostScript). The presumed availability of suitable processing software enables the recipes to be distributed in a form that is already marked up with the necessary *troff* typesetting codes. A reasonable proportion of the subscribers can process the

recipes through *ditroff* to obtain good quality laser-printed output.

The standard sets of commands for *troff* markup are collected in macro packages and the benefits of using some well-known set of standard macros for The USENET Cookbook will be apparent. Two widely distributed sets, much used for technical papers and memoranda, are called *ms* and *mm* but even these are not universally available, or have been so ‘enhanced’ by vendors as to be unrecognisable. The only near-universal set that could be found were the *man* macros, which are used for typesetting UNIX manual pages. Reid relates, somewhat glumly, that many of the manufacturers have ‘mucked about’ with this package too. Nonetheless, armed with some expertise in mounting new software packages, the more expert subscribers to The USENET Cookbook can install software to build up a database of recipes, together with a modified version of the *man* macros which enables the recipes to be typeset in a manner strongly reminiscent of a UNIX manual page. An example is shown in Figure 1.

In undertaking any experiment in electronic dissemination one has to keep in mind the highest and lowest common factors of hardware, software and computer knowledge among the subscriber base. The presence of a standard UNIX environment among USENET Cookbook recipients turned out to be wishful thinking. Many subscribers possess only a personal computer and a rudimentary dot matrix printer and for them a straightforward ASCII text version of the recipes has to be distributed. On the other hand a sizeable number of subscribers might have access to a PostScript laser-printer in the future (or perhaps a workstation running Display PostScript). One might then contemplate a distribution of the recipes as PostScript masters — though the very bulkiness of this format would require some form of text compression.

4. *EPodd* as a hybrid journal?

The experiences of BLEND and The USENET Cookbook convinced us that *EP-odd* should not start out as a fully electronic journal. Rather, we wanted to begin with the ‘hybrid’ form, already referred to, by using all possible technological aids to produce a traditional journal but leaving ourselves the option of phasing in various electronic methods for acquiring and disseminating information in the future.

We hoped that *EP-odd* could convey the latest research and current practice in EP but we knew that papers would be judged both on content and on evidence of care during printing and production. We decided that having hard-copy at various key stages was the only way of forcing ourselves, and our authors, to set high standards for content and appearance: it would be much easier to establish the reputation of some future fully electronic version if this could be built on the foundation of an existing reputable journal.

A second reason for favouring a hard-copy form is that the very use of the phrase ‘paper’ to describe a contribution to a journal indicates how deeply-ingrained are the expectations of the ultimate output medium. Although there have been several electronically distributed ‘papers’ on USENET bulletin boards, material in this format is usually regarded as interesting but ephemeral; there is no expectation that the articles have been scrutinised or refereed in any way. In cases where the subject matter has been of longer-term interest a mark of this greater worth is that the authors, or their colleagues, have arranged for traditional publication to take place [4, 13].

4.1. Input and output formats for *EPodd*

The decision about the output format of *EP-odd* prior to the hard copy stage was an easy one: from the outset we decided that the final ‘electronic’ form of all papers would be PostScript. This would enable material to be proofed on laser-printers at various stages during authorship,

CURRIED PEANUT CHICKEN

CHICKEN-PEANUT – Chicken in a spicy peanut sauce

I got this recipe from my cousin who lives in Amsterdam. She got it from a book of Jewish recipes from Curacao.

INGREDIENTS (Serves 4)

2 *Small frying chickens*, cut into serving-size pieces
60 ml *oil*
10ml *salt*, or less, to taste
15 ml *curry powder*, or more, to taste
1 *large onion*, sliced
1 *large green pepper*, cut into strips
1 large *tomato*, skinned and sliced
50 g *crunchy peanut butter*
60 ml *water*

PROCEDURE

- (1) Brown the chicken pieces in the oil. Stir the salt and curry powder into the drippings and cook, stirring, for one minute.
- (2) Add onion, pepper, and tomato. Cover and simmer five minutes. Add chicken pieces, cover and simmer 30 minutes or until tender.
- (3) Remove chicken pieces to a dish, or over rice. Blend peanut butter with water and stir into gravy. Heat to boiling while stirring constantly. Serve.

RATING

Difficulty: easy. *Time*: 45 minutes. *Precision*: no need to measure.

CONTRIBUTOR

Ed Gould
mt Xinu, Berkeley, California, USA
{ucbvax,decvax}!mtxinu!ed

Figure 1. A typeset article from *The USENET Cookbook*

editing and production. Material could be previewed in PostScript form on Sun workstations running the NeWS software, or on Display PostScript systems such as the one found on the NeXT computer. The final PostScript would be sent to a typesetting service equipped with some sort of PostScript typesetter and the bromides thus obtained would become the masters for the printing process. An archive of source code and PostScript files for all published papers would enable us to move into full electronic distribution at some future stage.

The input formats that we should be encouraging, and the range of media on which we could accept contributions, were more problematical to decide. At the one extreme we could devise a set of tags to capture the logical components of an *EP-odd* paper (title, abstract, section etc.) and ask the authors to use these. The drawback to this is that authors would not be able to

process their papers to see how they would actually look in the journal (we shall have more to say on this subject in section 4.5). At the other extreme the widespread availability of DTP software would, in principle, allow authors to lay out the pages of their papers for themselves. Previous experiences with this sort of software, and with ‘camera ready copy’ for various conference proceedings, convinced us that very few of our authors would have the graphics arts skills to achieve an acceptable page layout. We could all too easily envisage the immense amount of work generated for the publishers if they had to overhaul amateur page layouts done with DTP software.

After some deliberation we chose to standardise on three well-established text-processing systems namely, *troff* [14] T_EX [15] and Ventura Publisher[®]. The first two were chosen because of their ready availability among our prospective author base and because they could be customised with macro packages so as to mimic the journal style. The choice of Ventura Publisher was motivated by its availability at Chichester and the fact that, among DTP software, it is one of the better systems for producing longer documents and for allowing some of the logical structure of a document to be represented as a set of ‘tags’.

We have already noted that *troff* is distributed with most UNIX systems and that extra software to convert its output into PostScript is readily available. Don Knuth’s T_EX has also achieved considerable popularity, and a devoted band of disciples, in the years since its first introduction in 1978. T_EX is in the public domain and, again, various programs are available to convert its output into PostScript; unlike *troff*, it does not rely on having a UNIX environment at run-time.

Now the use of *troff* and T_EX for the production of journals is not new. As early as 1977, Lesk and Kernighan reported an experiment in using UNIX typesetting software on behalf of the American Physical Society’s journal *Physical Review Letters* [16], and more recently the *Australian Journal of Management* has adopted *ditroff* as its standard software for authoring and production of all its papers [17]. The American Mathematical Society has supported the use of T_EX for more than 10 years and a small number of other journals are prepared to accept papers authored with T_EX or L^AT_EX.

The novel factor in our plans was to distribute to our authors, on request, the same set of macros and tags, for each of the supported systems, that would be used in the final production of the journal. This would enable papers to be originated very much as they would finally appear. Our motive for doing this stemmed from the belief that one cannot divorce form and content in the world of electronic publishing. Line diagrams, equations, photographs and tables need to be seen at the right size and in the right positions; a particular choice of diagram or a paragraph in some non-standard font might alter the whole way in which a technical point was argued and presented. In short, authors would welcome the chance of being involved in the production cycle of their paper, of knowing its likely size and of judging whether its length was about right for the importance of the subject under discussion.

In asking John Wiley Ltd. to publish our journal we were very aware that text processing schemes such as T_EX and *troff*, having been written by computer scientists, would not necessarily be the systems of choice if the publishers were given a free hand. And yet the success of our plans depended on the production staff being prepared to proof and copy-edit the papers using the same text processor that the author had already chosen. Fortunately the staff at the Chichester office had already used T_EX extensively and had some limited experience with Ventura Publisher. A senior member of the production staff was willing to learn about *troff* and UNIX and to supervise all the later phases of producing the journal from the PostScript master files.

In the pilot issue of *EP-odd* we wrote an article about these arrangements [18] stressing the care that needed to be taken in preparing manuscripts and making plain the amount of time and energy that the publishers would be expending in allowing the journal to be created in this way.

4.2. Macros for the *EP-odd* style

The basic text formatting commands in *troff* are very much like a low-level assembly language. A command such as `.sp` denotes that a line space is to be inserted at that point, `.ft B` denotes a change into a bold font, `.ce` denotes that the next line is to be centred and so on. The fact that these instructions are so concerned with the appearance of the final page, and how it is to be achieved, gives rise to the term *procedural markup*.

A key advantage of making the markup be visible is that groups of commands for spacing, indenting, font change etc. can then be gathered together into macros. Quite apart from grouping together the typesetting functions, these macros take on an extra aura of abstraction by indicating something of the logical structure of the document. An example of this is shown in Figure 2 where the source text for a hypothetical front page of a paper has been marked up in a *troff* macro style. It is easy to guess the meanings of the two-letter codes for the macros: `.TL` is the title, `.AU` are the authors' names, `.AI` is the institution where the authors work, `.AB` marks the start of the abstract and `.AE` marks its end. Figure 3 shows how this same markup, when processed through *troff*, can produce very different effects, depending on the details of the macro package in use. Figure 3a shows the effect with the widely distributed *ms* set of macros whereas Figure 3b shows the effect with the *ep* macros which have been developed specially for *EP-odd*.

```
.TL
An Object-Oriented Methodology for Quality Control in the
Production of Sans Serif Fonts
.AU
G. BUZZWORD AND NEIL E. CONTENT-FREE
.AI
Institute for Object-Oriented Typography
42 Baskerville Boulevard
HAMBURGEFONSTIV
Korinna
.AB
In this paper a structured conceptual design is propounded,
leading to object-oriented virtual schemata for the
representation of sans serif fonts as Stieltjes-Tartaglia
intersections of various smooth structures in fake 4-space.
The prospects for improved rendering of stems and counters are
discussed.
.AE
```

Figure 2: Markup for title page of a paper using *troff*

The *ep* macros have undergone field trials with a few of our pioneer authors and have been distributed either by e-mail or on floppy disc. The similarity of the macros to those in the well-known *ms* set ensures that authors find the transition relatively easy.

The style of markup in $\text{T}_{\text{E}}\text{X}$ is similar in spirit to that employed in *troff*. Once again the basic typesetting primitives can be grouped together into macros. The main cosmetic difference is that markup for $\text{T}_{\text{E}}\text{X}$ is all of the ‘in-line’ variety which is to say that typesetting commands,

beginning with the `\` symbol can appear anywhere in the line and are not restricted to being only two characters long. An initial macro configuration is distributed with $\text{T}_{\text{E}}\text{X}$ itself and the resulting system is known as ‘Plain $\text{T}_{\text{E}}\text{X}$ ’. A more ambitious set of $\text{T}_{\text{E}}\text{X}$ macros has been developed at Chichester, specifically for use with *EP-odd*, and these have been used to produce four of the papers that have appeared so far. They have not been widely distributed among authors, mainly because of the increasing popularity of a more heavily configured version of $\text{T}_{\text{E}}\text{X}$, called $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ [19], which makes available a number of standard document styles for papers, books and so on. In response to requests we are now in the process of setting up a document style for *EP-odd* under $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

The philosophy of allowing procedural markup to describe layout effects has been followed through in $\text{T}_{\text{E}}\text{X}$ by building in facilities for mathematical equations and for tabular layout. The *troff* suite goes one step further by providing a range of pre-processors, each of which invents its own small language for describing the layout of specialised material. Thus we find *tbl* (for tabular material) *eqn* (for equations) *pic* (for line diagrams) and so on. These extra tools greatly enhance the range of material that can be typeset.

4.3. *EP-odd* by DTP?

Much of the software in DTP works on the basis of ‘what you see is what you get’ (WYSIWYG) and is largely concerned with the detailed appearance of a page rather than its abstract or logical structure. This sort of attention to layout detail is essential for advertising copy and for publications such as *Punch*, or *The New Yorker*, where cartoons may appear at various places in a page and in a wide variety of orientations. Text may be required to flow around the outline of cartoons or sketches and the text content of a page may have to be edited so that it fits exactly into the allocated page area. Such fine control over page layout is not necessary in an academic journal: the page lengths and widths are always standard and each page in a given article is just one of many rather than being a work of art in its own right.

On the other hand we were clearly going to be asked to support at least one DTP method for originating papers. We chose Ventura Publisher over its rival, Pagemaker[®], not only because of its availability at Chichester but also because the facilities for handling formats and document styles seemed suited to longer documents and journal papers. Significantly, perhaps, no paper has yet been submitted in this form though the Ventura tags are all prepared. These tags are not easy to disseminate via electronic mail because they are in a binary format; authors wishing to use them have to request a set on an IBM 5¼ inch floppy disc. Ventura allows material from a range of other DTP packages to be incorporated into a document as well as having a facility for importing Encapsulated PostScript (see next section).

4.4. Encapsulated PostScript

As we have already noted, both $\text{T}_{\text{E}}\text{X}$ and *troff* have facilities, either built-in or made available by preprocessors, for setting out mathematics, diagrams and tables. However, these simple descriptive schemes cannot cope with complex illustrations such as photographs, half-tones and pattern-filled diagrams. This sort of material can be generated from software such as MacDraw[®] or Adobe Illustrator[®] or may take the form of bitmaps obtained by scanning in photographs, line diagrams etc. and converting them to PostScript ‘image’ format. An increasing number of authors are choosing to send their illustrations to us in this form. Some examples of the sort of material involved are shown in Figure 4.

An Object-Oriented Methodology for Quality Control in the Production of Sans Serif Fonts

G. BUZZWORD AND NEIL E. CONTENT-FREE

Institute for Object-Oriented Typography
42 Baskerville Boulevard
HAMBURGEFONSTIV
Korinna

ABSTRACT

In this paper a structured conceptual design is propounded, leading to object-oriented virtual schemata for the representation of sans serif fonts as Stieltjes-Tartaglia intersections of various smooth structures in fake 4-space. The prospects for improved rendering of stems and counters are discussed.

An Object-Oriented Methodology for Quality Control in the Production of Sans Serif Fonts

G. BUZZWORD AND NEIL E. CONTENT-FREE

Institute for Object-Oriented Typography
42 Baskerville Boulevard
HAMBURGEFONSTIV
Korinna

SUMMARY

In this paper a structured conceptual design is propounded, leading to object-oriented virtual schemata for the representation of sans serif fonts as Stieltjes-Tartaglia intersections of various smooth structures in fake 4-space. The prospects for improved rendering of stems and counters are discussed.

Figure 3: Source text of Figure 2 typeset with *troff* using

- (a) the *ms* macros
- (b) the *ep* macros



(a)



(b)



(c)

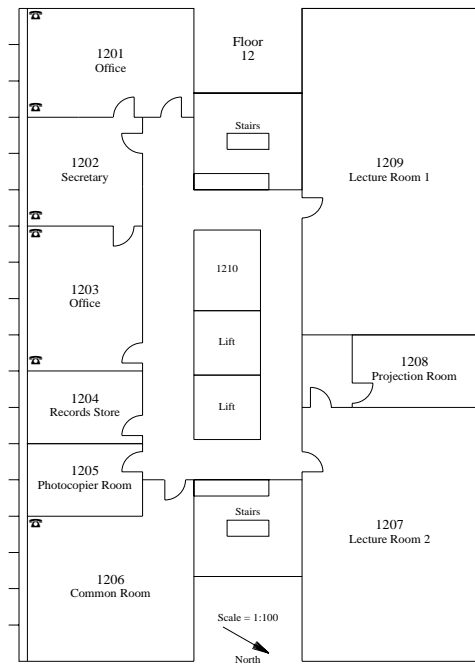
```

guide
Quit New Down/Up Save Block-edit Author
rm, rmdir: commands that remove (unlink) files
syntax: rm [-fri] file ...
Example
DESCRIPTION
Rm removes the entries for one or more files from a
directory. If an entry was the last link to the file, the
file is destroyed. This means that ... More. Removal of
a file requires write permission in its directory; the
permissions on the file itself are immaterial.

If a file has no write permission and the standard input

```

(d)



(e)

Figure 4: Examples of Encapsulated PostScript material

- (a), (b) the authors
- (c) crest of the British Computer Society
- (d) screen dump from a SUN workstation
- (e) line diagram of floor plan

Once the PostScript has been obtained there still remains the problem of how to incorporate it into documents whose ultimate output form will be PostScript but whose source format uses very different set of abstractions. Essentially, all that can be done is to define start and end markers at the source level, which denote where the PostScript is to be inserted, and to devise some means of telling the text processing software what the overall size of the diagram will be. The PostScript insert can then be passed on, unaltered, to the back-end device-driver module which merges the insert into the PostScript already produced from the surrounding material. These effects are provided by software such as *psfig* [20] which allows PostScript inserts to be included in \TeX and *troff* source code. The markers used by *psfig* are `.F+` and `.F-` and between these markers a small set of commands is allowed for indicating such things as the height and width of the diagram. To reinforce this information there must be a `BoundingBox` comment at the head of the PostScript file.

Unfortunately, the care that has to be exercised with the inserted PostScript is not limited to defining the `BoundingBox` correctly. The PostScript interpreter has a complicated internal state-vector with various global variables and matrices which can induce translations, size changes and transformations on the image. If PostScript from some source is mindlessly inserted into a *troff* or \TeX document with *psfig* it is all too common to find that it has shrunk to the size of a microdot or expanded to the size of a barn door. In an attempt to control and contain these effects there is a set of standards for what is now called Encapsulated PostScript [21] which, if carefully followed, enable PostScript to be transferred from document to document.

For the most part PostScript inserts will consist of complex graphics or scanned material that has been converted to PostScript bitmap form. Some examples are shown in Figure 5.

4.5. Why not use SGML for *EP-odd*?

The idea of separating the logical structure of a document from its physical appearance found early expression in systems such as Reid's *Scribe* [22]. In recent years a major international effort in this direction has defined a notation called SGML [23] (which is an acronym for **Standard Generalised Markup Language**). SGML is often, mistakenly, thought to be a fixed set of markup tags but it is in fact a *metanotation* which provides a syntax in which markup schemes for a wide variety of documents can be generated [24]. In this respect an SGML parser behaves like a compiler-compiler; the form of the desired markup tags is specified in SGML metanotation and the SGML parser-generator produces a 'compiler' which will 'compile' any document marked up with the defined tags. (The fact that the SGML parser itself may have been written using a compiler-compiler such as YACC adds nicely to the confusion.) Using SGML notation, it is perfectly possible to devise a Document Type Definition to specify the structure and sequencing of the various components in an *EP-odd* paper (title, authors' names, authors' affiliations and so on). An example of how the paper of Figure 2 might be marked up with a hypothetical set of SGML-defined tags is shown in Figure 5.

```

<front><title>
An Object-Oriented Methodology for Quality Control in the
Production of Sans Serif Fonts
<authors>G. Buzzword, Neil E. Content-Free
<instit> Institute for Object-Oriented Typography
42 Baskerville Boulevard
HAMBURGEFONSTIV
Korinna
<abstract>
In this paper a structured conceptual design is propounded, leading
to object-oriented virtual schemata for the representation of sans
serif fonts as Stieltjes-Tartaglia intersections of various
smooth structures in fake 4-space. The prospects for improved
rendering of stems and counters are discussed.
</abstract></front>

```

Figure 5: Example of SGML input document for the paper of Figures 2 and 3

On the face of it an abstract markup scheme is exactly what publishers need. Authors can create their books and papers on a word processor and can insert the appropriate tags denoting the boundaries of sections, chapters and so on. The processing of these abstract tags, to produce a typeset version of the document, would normally be done by the publisher rather than the author.

The problem with this approach is that it divorces structure and appearance so totally. SGML schemes can define the *syntax* of how document structure shall be denoted, in very much the same way as the syntax of a programming language can be specified. But, as everyone knows, it is the *semantics* of what this notation does, when it is ‘executed’ in some sense, that is the hard thing to tie down and control. In our case the desired semantic interpretation of the markup leads to a typeset form of the paper and we want our authors to be able to do this processing for themselves. However, for authors to convert SGML-based tags into laser-printed proofs they need a parser for the particular document structure and, moreover, one with suitable ‘code-generator’ modules to translate the SGML-type tags, and the text which they enclose, into a stream of commands for some existing typesetting system such as T_EX, or *troff*, or perhaps PostScript.

Clearly, if output appearance is the desired effect then an SGML approach adds extra layers of software processing (for the creation and use of the document parser) onto the front of existing text-processing schemes. Its proponents would argue that this leaves the author and publisher free to adopt different text processors to follow the SGML parsing stage but then there is no guarantee that the typeset effects from these different programs will resemble each other. We could see no reason to adopt SGML when so many of our authors can proof their papers by direct use of existing text processors and when our publishers were, in any case, happy to use exactly the same software as the authors. Although we do not use SGML as a precursor to the typesetting of *EP-odd* there may be some use for it in future if we need to annotate the published papers with tags for indexing or for database and hypertext access.

5. Producing an issue of *EP-odd*

5.1. Submission of papers

The rules for submitting a paper to *EP-odd* are very clear; papers are to be submitted in hard copy form, in triplicate, just as for a traditional journal. For all the reasons covered in section 2.1 we do not accept initial submission of a paper to *EP-odd* by electronic mail. When the papers arrive, either at Nottingham or at Palo Alto, acknowledgements are sent back to the author; one of the hard copies of the paper is kept on file and the two other hard copies are sent off to the chosen referees, together with the standard documentation asking for their comments. It is only at this stage that any degree of electronic intervention enters into the pre-publication phase; on the referees' report forms it is indicated that the reports may be mailed electronically to the editor, provided that these are confirmed with hard copy later on. The author is also sent a confirmatory postcard and is required to fill in a Transmission Form to give some indication of how the paper will be submitted if it is eventually accepted for publication. Revised versions of papers are gathered together at Nottingham for initial copy-editing before being forwarded to Chichester.

The approved communication media for accepted papers are:

- Electronic mail
- IMB 5¼ inch high-density floppy disc
- Macintosh 3½ inch disc
- Sun magnetic tape cartridge
- Typesetter bromides as camera-ready copy
- Straightforward typescript

Authors are strongly encouraged to submit their papers in one of the machine readable forms, having used an approved authoring system with the appropriate set of macros or tags. The use of camera-ready copy either as bromides, or in the 'electronic' form of PostScript master files, is strongly discouraged. So far the first two transmission methods have proved to be the most popular and authors have generally been happy to use one of the supported text-processing systems. The last-named method —traditional typescript—has not found favour with any author so far nor has it been necessary to re-key any material.

Having arrived at Nottingham the papers are transferred to a Sun 3/160 fileserver which is largely devoted to storing an archive of accepted papers (in source form) and referee reports on all papers. The editor-in-chief can then proof the paper using the appropriate set of journal macros. By the addition of a simple macro call at the start of each paper it is possible to ensure that the dates of submission and revision appear at the bottom of the first page and that appropriate running heads, containing short paper title and authors' names, appear at the top of all subsequent pages. It is at this stage, too, that the editor can assert far more control than would be the case in a traditional journal by doing some initial copy-editing and repagination on the source form of the paper. This preliminary page-proof stage has the bonus that the pages are very close to the form in which they will finally appear in the journal making it easier to spot 'typos' incorrect

heading styles, diagrams which exceed the measure and many other problems. Most important of all the detailed pagination of each issue can be planned and its total number of pages estimated.

Once this initial editing has been accomplished papers are sent to the Wiley production staff at Chichester on high-density IBM 5¼ inch floppy discs (soon to be superseded by Sun magnetic tape cartridges).

5.2. Final production stages of the journal

The main job carried out at Chichester is final copy-editing and careful repagination using the same text processing system that the authors have already used for preparing the paper. The source text of the papers, sent down from Nottingham on floppy discs, is read in and stored on an 8 MHz PC-AT compatible computer: this machine has part of its disc allocated to a UNIX file system and the *troff* package, together with its suite of pre-processors for tables, equations, line diagrams can then be run under Microport UNIX. Alternatively, the machine can be booted up from an MS-DOS file system and this is used for running MicroT_EX and Ventura Publisher. Facilities exist for transferring files between the two file systems.

It has become apparent, after producing four issues of the journal, that the speed of this machine leaves much to be desired when processing lengthy and complex papers. Within the next few months all the work will be transferred to a Sun 386 system which will be connected via a LocalTalk network to other Macintosh and IBM PC computers.

Once all the material is available for a particular issue, including the editorial, the papers are then processed through the appropriate text formatting systems to produce laser-printed proofs, which are sent to authors and editors for correction in the usual way. Once the final corrections have been made the altered source documents are processed again and the PostScript output thus obtained is saved in MS-DOS files. These are sent off, again on IBM 5¼ inch floppy discs, to a typesetting bureau, which processes the PostScript into bromide output. The bromides from this process are returned to the production department at Chichester and final pasting-up is carried out for illustrations that are not available in PostScript form: the finished masters are then sent off to the printing works.

5.3. Practice and Experience to date

At the time of writing about 40 papers have been submitted to *EP-odd* and, of these, 12 have appeared in the journal (including the pilot issue). It is evident that authors appreciate the ability of submitting and revising their papers in a familiar text processing system; 11 of the 12 papers so far accepted have used either *troff* or T_EX.

Use of e-mail

From the author's viewpoint the support of named text-processing systems for creating *EP-odd* papers seems to have been a success. However, from the editors' viewpoint, electronic mail is an equally vital ingredient in running the journal the way we do. All of the editorial board, with one exception, are contactable in this way and receive occasional updates on impending issues of the journal, usually accompanied by exhortations to commission more papers and to

solicit more contributions from the EP industry. Referees frequently use e-mail for sending preliminary reports on papers and we, in turn, use it to chase up referees who are late in submitting reports. The greatest benefit of all is that the production and editorial staff at Chichester are contactable by e-mail and day-to-day problems can be put right very quickly. The only drawback is the tendency to rely too heavily on this high-speed communication. It is all too easy to rely on referees who are on e-mail and to neglect those who are not; to forget about editorial board members who are not on e-mail and to indulge in unpopular brinkmanship by using e-mail to inform the production staff of last-minute alterations.

Submission methods

The papers we have received so far have arrived at Nottingham either on floppy discs, magnetic cartridges or by e-mail. A high-density floppy disc holds about 1.2 Mbytes, which is entirely adequate for the source text of a paper. The problems begin when papers include several diagrams as Encapsulated PostScript, and particularly if this takes the form of bit images. An illustration made up from a bitwise dump of a Sun workstation screen can easily generate 150 Kbytes of PostScript and this rises to well over 1 Mbyte if colour is required. For these reasons the magnetic tape cartridge is becoming more popular as a transmission medium because of the higher storage capacity it allows.

Despite the intrinsic unsuitability of e-mail for transmitting long documents a surprising number of authors ask to use it for sending the final versions of their papers. We have already alluded to the problems encountered when mailer software somewhere in the network decides to impose a limit on the size an individual ‘letter’. On one memorable occasion an author had broken up his paper into small chunks to circumvent such a restriction but had failed to realise that the portions of the paper would not necessarily arrive at Nottingham in the order that they were sent. The luckless editor-in-chief was faced with the NP-complete problem of joining together the portions of the paper into the uniquely correct serial order. Authors asking to submit their paper in this way are now entreated to put the title of the paper, and a sequence number, in the header of all e-mail messages.

Use of macro sets

The recognition that authors wanted some control over the form of their manuscript as well as its logical structure led us to develop and distribute sets of macros for imposing the *EP-odd* house-style and much debugging and upgrading has occurred — particularly with the *troff* macro set — as a result of these trials.

However, some authors still use the standard version of their chosen text formatter (usually the article style with LaTeX, or the *ms* and *mm* macros with *troff*). A minor problem here is that the default page area in these systems is close to A4 size and pleas from referees and editors to cut a paper down to 12 pages was all too often interpreted as 12 pages of A4 rather than 12 pages of the A5 format that the journal uses.

A more serious difficulty arises when authors set out pieces of program, tabular material, line diagrams or PostScript inserts so that they occupy the wider measure appropriate to A4 size paper. Adjusting a piece of displayed program code down to A5 size requires a knowledge, for

each language, of whether over-length lines can be wrapped around onto the next line or whether some sort of continuation marker is needed. In the case of line diagrams and tables it is necessary to know how to shrink these in each of the specialised descriptive languages that may have been used. Encapsulated PostScript, can often be shrunk by appropriate use of the `height` and `width` commands of *psfig*, but there are endless possibilities for this sizing to clash in some way with deeply-hidden parameters in the PostScript itself. As ever, difficult cases are invariably referred back to the authors for further action.

The *ep* macros for use with *troff* have been produced from a set developed by Kernighan and Lesk to model the style of the journal *Software Practice and Experience*. They incorporate macros of considerable complexity, using the *troff* diversion mechanism, which attempt to balance the length of pages, to eliminate ‘widows’ and ‘orphans’, and to give flexibility in placing diagrams within pages. The experience so far with authors’ attempts to use these facilities has been uniformly disastrous and it is distressing to note the considerable efforts that have been expended on pagination and diagram placement, only to have the production staff do a much better job by hand. The supposedly more skilled efforts of the editor-in-chief, at the pre-publication editing stage, have met with scarcely better results. For these reasons we have now disabled these features in the macro set distributed to authors.

To be fair the problem is an incredibly difficult one to solve in its full generality, particularly for papers where the ratio of diagrams to text is high and it is debatable whether any automated system will ever be totally satisfactory, though work by Kernighan and van Wyk [25] and by Rogers [26] seems to have gone some way towards solving the problem.

Another set of problems may arise when authors have supplied exotic macros of their own, at the head of their source files, to supplement the standard macro set. This is usually done to achieve some peculiar effect and in most cases they work satisfactorily. However, it can be extraordinarily time-consuming to investigate the odd cases when these macros work for the author but do not work at Nottingham or Chichester. This is just one example of the need for a thorough knowledge of the supported text processing packages, by editors and production staff, before troubleshooting of this sort can be undertaken.

6. Alternative publication methods

It is still early days for *EP-odd* but we are encouraged by the positive response to the journal so far. For the present we are content to stick with the ‘hybrid journal’ idea and to enlist all possible electronic aids in support of a journal whose definitive version still appears on paper. However, a full database of published papers in source and PostScript is gradually going to accumulate and we should review very briefly the prospects for alternative ways of disseminating *EP-odd*.

***EP-odd* on CD-ROM or Videodisc**

CD-ROM technology involves the adaptation of the digital Compact Disc (CD) format to act as a read-only data storage mechanism for computers. The high storage capacity of CD has made it popular for disseminating bulky publications such as indexes, parts catalogues and technical manuals. We foresee that CD-ROM may turn out to be a valuable adjunct to conventional

publication for a wide variety of journals. One CD-ROM disc could contain several issues of a journal and computers equipped with CD-ROM readers (perhaps in public libraries) would be able to extract articles from a journal and arrange for them to be laser-printed on demand.

Videodisc is another medium which offers considerable potential as a means of publication. The recording method allows either analogue or digital information to be stored and a high degree of interactive browsing is possible if a videodisc is accessed through a suitable front-end computer system. Thus, BBC Television's experiment with this medium — called 'The Domesday Book' — uses a BBC Microcomputer to access a specially modified video disc player; this enables users to browse through a disc containing geographical, demographical and sociological information for the British Isles. Another recent videodisc project supervised by one of us (Beach) aims to produce an issue of the SigGRAPH journal on videodisc, complete with full-colour graphics for the illustrations and allowing animated graphic sequences to be displayed.

A fully electronic *EPodd*?

The barriers to disseminating *EP-odd* as a fully electronic journal are not only technological but also financial and legal. The existing networks are capable, in principle, of carrying source or PostScript versions of papers — though the bulkiness of PostScript, and the present charging structure for packet switch telecommunication services, does not exactly encourage this.

To realise the full potential of electronic publishing one needs to be able to transmit large files using a file transfer protocol and, possibly, to log on to some remote machine for a terminal session in order to browse through indexes and past papers. When high-bandwidth fibre-optic cable is commonplace the overall cost of data transmission might be low enough that subscribers would prefer to receive a copy of a paper in this way, rather than photocopying the article or obtaining a copy from some CD-ROM or videodisc version in a library.

Copyright problems

The size of journals and books is determined by a delicate compromise between costs of production and distribution. It is not economically worthwhile to put costly covers and fancy bindings onto a journal issue consisting of only half a dozen pages. At the other extreme the costs for producing a journal issue of one thousand pages can only be contemplated for much-needed items such as telephone directories. This delicate balance was first disturbed by the invention of the photocopier and at one time it must have seemed that these devices would deal a death-blow to the small specialist journal. However, journals have managed to survive, despite severe cutbacks on library budgets, and it seems clear that the irreducible need for properly bound and presented copies of journals will ensure a modest but stable market for some time to come. What the photocopier has done is to remove an extra tier of subscribers who at one time might have purchased a subscription as being the only method of acquiring the papers they needed. It remains the case that many photocopies are in actual or technical breach of copyright but publishers have the small consolation of knowing that there will be an inevitable degradation of quality when a copy is taken, and this fact alone preserves the integrity of the original journal as being the only genuine source of information. The dissemination of a fully electronic version of a journal poses much more serious copyright problems and it may be useful here to draw an analogy

with experiences in the music industry regarding recordings, on CD or vinyl, and the copying of these onto cassette tape.

In the music industry the parallel event to the impact of photocopiers on journal publication has been the invention of the cassette recorder, which has led to sizeable loss of revenue for the record companies. Piracy takes place on an enormous scale and attempts by the industry to have a levy imposed on blank tapes have met with little success. The small grain of comfort is the inevitable loss of quality whenever a tape recording is made and the fact that an investment in a digital CD system will give excellent reproduction that tape cannot match. But what has frightened the recording industry is the arrival of Digital Audio Tape (DAT) which, in principle, allows the digital information on Compact Disc to be transferred as a perfect copy, with no degradation and complete with all error correcting codes, onto digital tape. Frantic efforts are under way to ensure that the encoding systems on CD and DAT are as incompatible as possible and dark threats have been made that DAT recorders will not be released in any quantity unless there are foolproof schemes for preventing this 'perfect piracy'.

In the same sort of way an electronic copy of a journal, as PostScript, say, is indistinguishable from the original. The PostScript masters, if they were made available too early or too readily could easily be used to produce fully bound 'pirate' copies of any journal for which there is sufficient demand. Faced with this fact, and given the difficulties of billing subscribers to an electronic journal on a per-article basis it may well be that publishers choose the intermediate and more controllable media of CD-ROM or Videodisc before embarking on full electronic dissemination.

7. Acknowledgements

The successful launch of *EP-odd* owes much to many of our colleagues. We thank particularly Mark Bide, James Brind, David Evans, Geeti Granger Gaynor Redvers-Mutton and Nicola Smith.

References

1. D F Brailsford, "Electronic Publishing and Computer Science," *Electronic Publishing—Origination, Dissemination and Design*, pp. 13-21, 1988. Pilot Issue
2. D F Brailsford and R J Beach, "Editorial," *Electronic Publishing—Origination, Dissemination and Design*, pp. 1-3, 1988. Pilot Issue
3. Andrew S Tanenbaum, *Computer Networks (2nd Edn.)*, pp. 35-43, Prentice-Hall, Englewood Cliffs NJ, 1988.
4. Brian Reid, "PostScript and InterPress: a comparison," *Documentation Graphics—SIGGRAPH '86 Tutorial Course Notes No.29*, pp. 49-63, 1986. Also on USENET 'Lasert-lovers' distribution April 1985
5. Adobe Systems Inc, *PostScript Language Reference manual*, Addison-Wesley, Reading, Massachusetts, 1985.

6. Ted Nelson, "Replacing the printed word: a complete literary system," in *Information Processing* 80, pp. 1013-1023, 1980.
7. T Sheridan, J Senders, N Moray, J Stoklosa, J Guillaume, and D Makepeace, *Experimentation with a multi-disciplinary teleconference and electronic journal on "Mental Workload"*, 1981. Report (unpublished) to National Science Foundation
8. M Turoff and S R Hiltz, "The Electronic Journal: a progress report," *Journal of the American Society for Information Science*, vol. 33, no. 4, pp. 195–202.
9. B Shackel, "The BLEND system programme for the study of some electronic journals," *Computer Journal*, vol. 25, no. 2, pp. 161-168, 1982.
10. B Shackel, D J Pullinger, T I Maude, and W P Dodd, "The BLEND-LINC project on "Electronic Journals" after two years," *Computer Journal*, vol. 26, no. 3, pp. 247-254, 1983.
11. W P Dodd, *Convergent Publication or the Hybrid Journal: paper plus telecommunications*, May 1989. Paper submitted to *EP-odd*
12. Brian K Reid, "The USENET cookbook— an experiment in electronic publishing," *Electronic Publishing — Origination, Dissemination and Design*, vol. 1, no. 1, pp. 55-76, April 1988.
13. C. Bigelow, "Typeface Protection," *PostScript Language Journal*, vol. 1, p. 28–31, March 1987. Also on USENET 'Laser-lovers' distribution April 1985
14. J. F. Ossanna, "NROFF/TROFF User's Manual," Bell Laboratories: Computing Science Technical Report No. 54, April 1977.
15. D. E. Knuth, *TEX and METAFONT: New Directions in Typesetting*, Digital Press and the American Mathematical Society, Bedford Mass. and Providence R.I., 1979.
16. M E Lesk and B W Kernighan, "Computer Typesetting of Technical Journals on UNIX," *Proc. AFIPS NCC*, vol. 46, pp. 879-888, 1977.
17. Robert Marks, "On Redesigning an Academic Journal," *Australian Journal of Management*, vol. 12, no. 2, pp. 149-157, December 1987.
18. D F Brailsford and R J Beach, "Authoring Systems for *EP-odd*," *Electronic Publishing — Origination, Dissemination and Design*, pp. 4-9, 1988. Pilot Issue
19. Leslie Lamport, *L^AT_EX: A Document Preparation System*, Addison-Wesley, Reading, Mass., 1986.
20. Ned Batchelder and Trevor Darrell, *Psf_{ig}—A Ditroff Preprocessor for PostScript files*. Computer and Information Science Dept. University of Pennsylvania. Internal Report
21. Adobe Systems Inc, *Encapsulated PostScript File Format*, March 1987. (EPSF Version 1.2)
22. B. K. Reid, *Scribe: A Document Specification Language and its Compiler*, 1980. Technical Report CMU-CS-81-100 : Carnegie-Mellon University
23. ISO Standard 8879. Information processing, *Standard Generalised Markup Language (SGML)*, British Standards Institute, 1986.

24. David Barron, “Why use SGML?,” *Electronic Publishing—Origination, Dissemination and Design*, vol. 2, no. 1, pp. 3-24, 1989.
25. B. W. Kernighan and C. J. Van Wyk, “Page Makeup by Postprocessing Text Formatter Output,” *Computing Systems*, vol. 2, no. 2, pp. 103-132, Spring 1989.
26. D. F. Rogers, “A page make-up challenge,” *TUGboat*, vol. 9, no. 3, pp. 292-3, 1988.