The Inside Story of Life at Wiley with SGML, LaTeX and Acrobat*

Geeti Granger

John Wiley & Sons Ltd,
Baffins Lane, Chichester, W. Sussex PO19 IUD, England

1 Introduction

As a brief introduction I should say that John Wiley & Sons is a scientific, technical and medical publisher. It is an independent, American family-owned company that was established in 1807, with subsidiaries in Europe, Canada, Australia and Singapore. The European subsidiary opened in London in 1960 and moved to Chichester in 1967 (if folklore is to be believed this was so that the then Managing Director could more easily pursue his love of sailing!).

We publish books, including looseleaf and encyclopaedias, and journals, and most recently electronic versions of some of our printed products. In the future the electronic component of our publishing programme is bound to include products that are only available electronically.

2 Setting the Scene

Now to the topic in hand — Portable Documents: Acrobat, SGML and TeX. Our association with TeX dates back to 1984 when we made the significant decision to install an in-house system for text editing and composition. It was the only software available that wasn't proprietary, which stood a chance of coping with the complex mathematical material we had to set.

As a company we have monitored the progress of SGML since 1985, but have only recently used it in earnest. Our first project is a 5000 page encyclopaedia about Inorganic Chemistry. We rarely get the opportunity to dip our toes in the water — it's straight in at the deep-end! Having said this, we do have a set of generic codes that has been used for a number of years, and everyone is well aware of the principles involved and the value of this approach to coding data.

Adobe Acrobat was launched in June 1993. Our experience of this software dates back a little further than this, because of our links with Professor David Brailsford and the Electronic Publishing Research Group at the University of Nottingham, and their work on the CAJUN (CD-ROM Acrobat Journals Using Networks) project, which we jointly sponsored with Chapman & Hall.

3 Complementary not Competitive

The first thing to make clear is that SGML, TeX and Acrobat do not compete with each other in any way. SGML is a method of tagging data in a system-independent way. TeX is one possible way of preparing this data for presentation on paper, while Acrobat is software capable of delivering data electronically for viewing on screen, or for committing to paper.

From our point of view the fundamental requirement for:

- capturing data
- processing data (text and graphics)
- delivering data (paper/disk/CD/Internet)

is to remain system independent for as long as possible.

SGML, TeX and Acrobat achieve this in their part of the whole process. PostScript provides the link that completes the chain.

4 SGML in Practice

To describe our experience with SGML I will use the *Encyclopedia of Inorganic Chemistry* as a case study. This encyclopaedia is an 8 volume set made up of 5000 large-format, double-column pages (more than 3 million words). The data consists of approximately 250 articles interspersed with 750 definitions and 750 cross-reference entries. The text was marked-up and captured using SGML, validated and preprocessed for typesetting. The floating elements (all 2300 figures, 8000 equations, 2000 structures, 1100 schemes and 900 tables) were prepared electronically and delivered as encapsulated PostScript files. Some 150 halftones, about a third of which are colour, complete the data set!

Despite the complex nature of this project, or maybe because of it, we were convinced that using SGML was the right approach. We had to be very sure because this decision presented us with many additional difficulties. Different considerations had to be made at all stages of the production process. (Manufacturing remained untouched.)

Initially, having established the probable requirement for an electronic version, there was the need to justify the use of SGML because of:

- the extra cost involved in data capture

*Reprint from the Annals of the UK TeX Users Group **Baskerville**, Volume 5.2, March 1995. Published with permission of both Baskerville editor and author. Presented at the UK TeX Users Group conference 'Portable Documents: Acrobat, SGML, and TeX', on 19 January 1995, London, England.

- the different working practices that had to be established
- the project management overhead
- the need to find new suppliers, and the risks that this involved for such a large, high profile project.

4.1 Production Considerations

This project had an external Managing Editor to commission and receive contributions before it became a live project for us. Once contributions started to arrive it very quickly became apparent that a project management team was needed if this project was to succeed. The initial steps had to be ones of project analysis, determining data flow, deciding who was responsible for what, and ensuring that a progress reporting system was established. It certainly seemed like a military operation at times.

Having made the decision to go with SGML and to ensure that all components were captured electronically we had to find a set of new suppliers. None of our regular suppliers could meet our specifications. Locating potential suppliers was the first hurdle, and then assessing their suitability was the next. Having done this we then had to draw them all together to establish who did what, and who was responsible for what. It had to be a team effort from start to finish and regular progress meetings involving representatives of all parties was the key to an ultimately successful project.

4.2 Problems Encountered

One of the first considerations was how on earth do we name the files? To ensure portability we set ourselves the restriction of the eight plus three DOS convention. It took some time but we achieved it in the end so you can now identify from the file name the type of text entry, the type of graphics and whether it is single or double column or landscape and its sequential placement within its type. When you consider the number of files involved, this was no mean feat.

Designing the DTD without all the material available is not the best way to start, but needs must. It meant that some amendments had to be made as the project progressed but none of them proved to be too significant.

Choosing Adobe typefaces, to avoid problems later on, meant that some compromises had to be made. Many people feel that the Adobe version of Times is not as elegant as some.

Also the quality of the typesetting, hyphenation and justification, interword spacing and overall page make-up is not as high as that normally achieved by a dedicated chemistry typesetter.

In addition to the above, we found a bug in Adobe Illustrator! Because the EPS files were being incorporated electronically the accuracy of the bounding-box coordinates was crucial. To cut a long story short they weren't accurate. We spent quite some time establishing the cause of the problem and then had to have a program written to resolve it.

This is not an exhaustive list but I think it will give you a feel for the practical issues involved. Having shared all this with you I should add that all of us involved in the original recommendations remain convinced that it was the right approach. In fact we are now processing two more projects in the same way!

5 LaTeX in Practice

We've done far too many projects in TeX (many in Plain, but a growing number in LaTeX) to select one as a case study. What I can do is very readily identify the production issues involved in using this software in a commercial environment.

5.1 Steps in the Process

Establishing ourselves as a forward-thinking, progressive company by developing in-house expertise has brought with it certain pressures. In the early days, not only did we have to learn how to use TeX, we also had to make it achieve typesetting standards expected of more sophisticated systems. Our colleagues could not see why they should accept lower standards from us — after all they were paying us (we operate a recharge system so that it doesn't distort the project costing when compared with externally processed projects).

Next came the requests for us to supply style files. Authors knew we used the same software as they did, and wanted to prepare their submission so it looked like the finished product. Some wanted to produce camera-ready copy. In principle this would seem a sensible idea; in fact our commissioning editors, especially those who handle a number of CRC projects, thought it was a brilliant idea. It would save them an immense amount of time and hassle.

Now, preparing style files for in-house use is one thing; preparing them for use by others is something else again. We have to work within strict time and cost constraints, and there are many occasions (dare I admit it?) when we have to resort to, shall we say, less than the most sophisticated way of achieving the required visual result!

When I have attended courses on TeX and have asked about writing style files the answer has often been along the lines of 'leave it to the professionals'. (I should say it's usually people who make their living in this way who give this response.) This may be fine if a) you can find and afford the professional; b) you don't need to support the file when it is in general use. In our experience the first is difficult to do and the second is an impossibility. The need to support style files cannot be ignored; once they have been provided, no matter on what pre-agreed conditions, queries will arise. It can be very time-consuming, as often queries are not restricted to the style file, but relate to the sytem being used. It can also take a while to establish the context of the query, resolve it and respond. To meet the expectation that we will support, customise at short notice, resolve technical issues, and communicate via e-mail (preferably responding within the hour) can be difficult, given the level of human resource available.

Once you've got over this initial stage, the practical issues involved in accepting LaTeX submissions can be many. Delivery is the first. Now that we have the ability to receive data electronically our authors cannot understand why we hesitate, and why we still insist on hard copy. Experience tells us that, without hard copy, it is difficult to be sure we have received the final version, and discovering this after a project has been processed is very costly, both in time and money. Any submission that circumvents a stage in the current administration process may drop through a hole and end up taking more time, rather than less, to reach publication. Consideration is being given to this issue, and there is no doubt that in the future electronic delivery will be an acceptable method of submission, but in the meantime everyone has to be patient.

Copy-editing remains a conventional process in the main, although experiments are taking place with copy-editing on disk. This issue is not resticted to LaTeX projects, but the rate of progress is dictated by the ability of our freelance copy-editors to provide this service.

Once you move on to the processing stage the first thing you have to do is find a supplier who is capable of actually processing in this software. This is easier said than done, because it is not considered to be cost-effective by most of our regular suppliers. However, as a result of our persistent requests, some can now provide this service, so we don't have to process all such submissions in-house.

From our own experience we know that producing page proofs is not always straightforward. Over the years we have struggled with amending style files to achieve the correct layout and controlling page make-up. Now that authors are submitting graphics on disk, as well as the text, we are faced with another set of problems. Portability of graphic formats is even more difficult to achieve. I think the number of answers to the question 'When is a PostScript file (or EPS file) not a portable PostScript file?' must be infinite. Even when the content of the file itself is OK, you can still be faced with problems in achieving the required size and position on the page.

Despite all these disadvantages our lives would not be the same without LaTeX, and when compared with processing in other software it can be a real joy! Our archive of projects coded in a form of TeX will be far easier to reuse than those processed in other software.

6 Acrobat at Arm's Length

Although we haven't used Acrobat on a live project in-house yet, we have been closely involved with the development of the EPodd CD. The CAJUN project has been running for well over a year and during this time the complete archive of volumes 1–6 has been converted to PDF, annotated to add PDFmarks and generally massaged into a suitable format for delivery on CD.

As always, the work involved in such a project is more than anticipated at the outset, but it has been an invaluable learning exercise. Being involved in the beta-testing of the software helps you appreciate just how much development work is required for a new piece of software, and although it currently has its limitations the future looks good. Version 2, which is due for release any day now, is much improved, and it is rewarding to see that many of the comments put forward by members of the team have been incorporated.

We are experimenting with small projects in-house to give us a deeper understanding of the practical advantages and limitations of Acrobat. It is easy to get caught up in the euphoria and hype that accompanies the release of a new product, and to overlook the day-to-day difficulties its rapid adoption might bring. Having said this, there is no doubt that it will have a place in our publishing procedures, and may be used in the production cycle for journal articles. Provided that the general administration can cope with the deviation from the norm, supplying author proofs in this way has its attractions. The fact that readers are now freely available and the PDF file can be read on any of the three main platforms is a real boon.

The use of Acrobat for delivering existing print products in an electronic form is one worth considering, especially now that it is possible to integrate it with project-specific software and the security issue has been addressed.

From an inter-company point of view the perceived use of Acrobat for distributing internal documents could again have its attractions. For this to be a real possibility it must be recognised that the use of such procedures is not an innate skill, and so the appropriate level of training and support must be available if it is to be successful.

7 Conclusion

The comments I have made and the case study I have described may leave you with a somewhat negative feeling. I wonder if I have emphasised the problems and not balanced these by identifying the plus points. To put this into context I should say that details of the advantages of any particular approach are usually more readily available, so I have tried to capture a more down-to-earth view.

In reality I am very enthusiastic about the use of SGML, TeX and Acrobat, but am also well aware of what their use in a productive environment can mean. I believe, as do several of my colleagues, that portability of documents is crucial to our ability to deliver data efficiently in a variety of forms, whether this be page-based, highly structured databases or tagged ASCII files. To this end we must be flexible in our approach, and must not be afraid of making investments now that may not bear fruit until some time in the future. This can be a very unnerving decision to make, and for one I am glad it isn't ultimately mine. While I can extol the virtues of a purist's technical approach, obtain the relevant costs and assess the schedule implications, I do not have the entrepreneurial skills required to know when a project is commercially viable (or worth taking a risk on). It is at this point I take my hat off to our commissioning editors, who have the responsibility for turning these experiments into profit for us to reinvest in the next Big Thing!