

Torsten Madsen

From manuscript to book:
Specifications for an integrated
publication production environment

Paper presented at the conference Computer Applications
and Quantitative Methods in Archaeology CAA92 – Aarhus
27th-29th March 1992

Appear as read with minor corrections to grammar and language.
As mentioned in the text it was not the intention that this paper should be
published. Looking at the actual development in publication production in
archaeology, it may now be of some historical interest.
Torsten Madsen 1-10-2003

The title of this paper is perhaps a little pompous. The more so, as I am not going to give exact specifications for a publication production line. What I am going to do is to discuss a number of issues related to the publication process. They are:

- * The practical problems we are currently faced with doing publications, and what we can expect and demand of authors in archaeology.
- * What kind of production types are adoptable to archaeological publications, and indeed what types we will be able to handle within archaeology at all.
- * And finally, do current hardware and software development point in directions, where stable, structured solutions for publication productions can be seen in a not too distant future.

I recall that Sebastian Rahtz had a paper at CAA 90 in Southampton called "Desk top publishing, no thanks!". It was never published, and neither will this paper be. I suspect that Sebastian's paper was motivated by his position as organiser of the Southampton conference, and by the task ahead of him as editor of the proceedings of the conference. At least that is the motive for me to have this paper on the program now, although I have already been faced with the problem for some years, due to the intensive book production taking place at Moesgård. You may already have noted that many of the new books on display here at CAA are Moesgård productions.

Recalling the title of Sebastian's paper I was tempted to call this paper for "Desktop publishing yes please!", as the solution I will argue does lie within that realm. It would, however, have been a little misleading, as I am not primarily pleading for desktop publishing. Rather I am pleading for a publication production that can be linked to the type of products we receive from archaeological authors, and at the same time is structured, flexible, and easy to use given the special kind of books that archaeologists produce. Should I have had another title along the lines of Sebastian's, it might have been "Structured publishing yes please!", a title that we probably both would be able to agree upon.

Let me start with a closer look at some of the current problems we are faced with. Although Moesgård is much larger than the average institution, and untypical have 50-60 PC's connected through a Novell LAN on an Ethernet basis, it is typical of Danish archaeology in the sense that computing is entirely PC-based. Despite the network there are no standards within the institution to be followed. On the hardware side most of the PC's are DOS compatible, but there are clusters of Macintosh's as well, some 15 in all, and indeed the machines at the graphics department, where most of the book production is funnelled through, are Macintosh's. On the software side there is an even greater diversity. For instance a large part of the word processing takes place on WordPerfect, while another large part is on either Word for Windows or Macintosh Word. Programs like Word for Dos, IBM Display Write, PC-write, Windows Write and Macintosh Write Now are also in use.

However, our publications are not limited to authors from Moesgård. In the edited books, most of the authors comes from outside. This means that we receive a variety of disks with files from even more word processor types, including some odd Danish ones like Dantekst and DSI Tekst. Our publication production programs have no import filters for these. We are pleading with all authors that in addition to the word processor files they should include an ASCII file on their disks. In many cases, however, the authors don't even know what ASCII is, and far less do they know how to produce ASCII files from their word processors.

A further complication is that many - in fact most - authors do not know how to use a word processor properly. Their minds are often completely blocked by the idea that they should be able to send a paper copy to the editor that looks as fancy as possible. Often they obtain this result by using "typewriter formatting methods" like applying scores of blanks and tabs.

Thus as an editor you have three different nuts to crack before you can get to work:

- 1 You have to read the files (disks) physically
- 2 You have to transform the text in the files to your own word processing platform. In most cases you can do this by way of import filters, but now and again you have to do some laborious editing of files that comes from odd word processors.
- 3 You have to clean up the text for all the odd things the authors have done in order to make the text look nice.

All this and indeed the whole set of relations between authors and editor spells CHAOS with capital letters. We must find ways to lessen this chaos, as right now it is threatening the whole concept of electronic integrated publishing. Basically, two things must happen: One is that the authors must become educated in using word-processors. It is untenable that they do not know the basic rules of using a word processor. They must be taught to format the text correctly, and they must learn how to import and export texts in the various formats that their word-processor support. The other thing is that a certain standard of word-processors has to be attained. Right now many of the big international word-processors are converging standards of at least their add-ons so that exchange of data is facilitated. We should carefully watch this tendency, and try to uproot those products that appear not to follow suit. It is increasingly unacceptable that word-processors are in use, which does not allow for data exchange of formatted text with other products.

Our current problems are however not limited to the author - editor relationship. There are also obvious problems in the editor - production relations as well. The initial editing is normally done on the printed version that the authors supply, and often based on this the authors are asked to make corrections even before an attempt to read their files has been made. However, once the text has been imported, the next step is to produce a type set version. At Moesgård - if it is the graphics department that has to run the production - the text is transferred to the Macintosh environment - normally by way of Microsoft Word, and is then type set in Quark Express. This actually locks the text within this limited environment.

The first proof after the text has entered Quark Express is a proof of the type set text only. The corrections following this proof have to be made in Quark Express on the Macintosh. It thus becomes an action with severe limitations of where, when and whom, and it turns out that we get bottle-neck problems, and a not at all satisfactory production, where authors and editor mark their corrections on printed copies, which then are delivered to persons who do all the corrections from one end to the other - where and when time can be found. This process is repeated for the second proof, with the final page set-up including tables and figures.

The locking of text in an interactive desktop publishing system like Quark Express raises the question whether this type of system is the right one to use, or whether we should prefer other types of systems, like TEX, where you run the typesetting in batch mode based on a mark up system, and thus have the correction of text fully freed from the typesetting production.

The question of choice of system, however, is not one that can be decided on this issue alone. There are other more important things to take into consideration.

First of all we should note that archaeological publications are rather complex and variable in their appearance. Most publications are in two columns and the numerous figures, and tables are interchangeable run in column or full page width. Further, figures and tables are often referred to in the text in bundles leading to a rather uneven placement in the final page layout. Even though you naturally try to follow general rules of how figures and tables should be placed on the pages, then in many cases you have to make exceptions to the rules, and producing a good layout of an archaeological book really calls for a professional eye and hand.

General mark up systems like TEX are best suited for those types of publishing, where you have a very strict, repeating structure, where every possible situation can be foreseen and dealt with in the programming of the procedures that defines the publication. Two things are prohibitive here. First of all we do not have the expertise within archaeology to do the necessary programming, and secondly it is very doubtful whether it would actually be possible to program an automatic set-up of an archaeological book that would appear satisfactorily to the eye.

If we were to use TEX for the typesetting, we would as far as I can see have to use the traditional manual mounting of the pages, where the text is typeset in a continuous stream, and then cut up and pasted onto mounting sheets together with tables and figures. This procedure is in fact just as quick as the on-screen mounting that takes place in for instance Quark Express, but our draughtsmen prefer the latter because it is more precise.

Thus my conclusion would be that a mark up system like TEX is excellent where a systematised, repeating set of data are to be published, as we have seen in the two previous papers. It does not, however, work in cases where complex, versatile layouts are called for. Thus we will either have to use manual mounting, or we will have to use a desktop program that allows for interactive, intuitive page layout. The question however is: do desktop publishing systems provide sufficiently good typesetting qualities and do they provide a sufficiently structured approach to publication production?

It is probably commonly known that desktop systems have a really bad reputation. They are considered incapable of producing quality typesetting. This certainly used to be the case, and probably still is the case with the many minor and inferior systems available on the market, but it is not true any more with the larger more prominent systems like Quark Express, Ventura, and Page Maker. Our draughtsmen in the graphics department, who have been very critical and very much against using desktop systems have completely, changed their minds now. They claim that it is very difficult to find anything to point fingers at in the typesetting capabilities of the three just mentioned programs.

So with respect to typesetting quality there should be nothing to hold us back from the professional desktop publishing systems, but what about the structuring capabilities. Are they sufficiently good, or are the systems still of the happy anarchistic type, where everything is possible, and nothing can be formalised? Well actually the systems have changed a lot, and very much for the better. Let me turn to Page Maker, as it is the system we work with on the IBM side of the fence, and as it is going to be used for the CAA92 proceedings.

Page Maker in its first version was a terribly unstructured system. You could do what ever you liked, and no restrictions could be put on your possibilities of interactively screw things up. Page Maker in the current version 4 has retained the full capabilities of intuitive setting up of pages, allowing a person, who knows nothing about the program, and nothing about typesetting to screw things up as usual. Page Maker version 4, however, do also contain advanced features for structured formatting of the text allowing a person who knows the program, and who knows something about typesetting to do a

professional job, and indeed to set things up for others with little knowledge to do an acceptable job.

The structuring is based on named styles, where you can define in detail how text with a certain name should be formatted. The formatting comprises all levels of the text. You can define upper and lower limits for spacing between letters and spacing between words. You can define leadings as a percentage of the letter height (based on a standard height or on the highest letter in the line), or you can define it as a fixed leading. You can have a complete control for different types of whores (French and whatever they are named). You can have an automatic hyphenation based on a combination of a glossary and an algorithm, and it will at the same time look at the total paragraph in an attempt to minimise violations of the rules you have given for spacing between letters and words. If you have defined space ahead of a paragraph, you can have this automatically removed if the paragraph happens to begin at the top of a page. And not least if you have different types of styles with different leadings you can define a base net to which a given style must adjust by adding space either before or after the paragraph. By using this facility with care you can automatically avoid the ugly lack of register between lines in columns.

As far as I can see, there is just as much structuring in Page Maker as there is in for instance TEX. The principal difference is that the programming of Page Maker with its menu driven definition of types is much easier to learn and handle than is the more traditional programming used in TEX. And then what I consider the most important feature of a system like Page Maker is that when the structuring of styles is not sufficient to give a good result, and that happens now and again, you have always got the possibility to intervene and stretch out things in an unstructured, intuitive way. It may be dirty to do so, but it is efficient. In TEX you have to keep on programming until you reach a satisfactory result, or give up and call it a day.

So with a system like Page Maker the typesetting is OK, and the structuring capabilities are OK, but what about integration and flexibility in relation to the surrounding world, from where data is coming, and with which communication is to be upheld. I mentioned the problems we have experienced with Quark Express in the graphics department, and may add that these have in fact caused much frustration and even anti-computer publishing notions. I do not know Quark Express and the Macintosh environment sufficiently to say if there is a satisfactory solution to be found. I do know Page Maker and the DOS environment, however, and I have been quite pleased to find that Page Maker has come a long way towards integration.

Starting to discuss integration is really to open up a lot of issues, but let me concentrate on two: the reading of foreign formats using import filters, and the use of standardised formats for data exchange on a page-formatted level.

It has been increasingly common for word-processors to support import (and to a certain degree) export of foreign formats. This of course is also true with the dedicated desktop publishing systems. The list of formats that Page Maker reads is quite impressive, and it is indeed a very helpful feature, when you receive contributions from numerous sources. However, Page Maker has taken the support of foreign imports a little longer than to the level of just reading the foreign format, and structure the import accordingly.

If you have a word-processor that make use of named styles like Word, Word for Windows, or the current version of Word Perfect, then Page Maker is making use of the names. If names met in an imported document do not exist in the Page Maker document, a style with the name and formatting found is created. If, however, the name already exist in Page Maker the format attached to the imported text is overwritten with the format defined for the name in Page Maker.

This is indeed very helpful. By agreeing to names of different types of text it is possible to obtain a full integration, where text received from different authors using different word processors, and having different actual formatting of their text, will be formatted as defined in Page Maker when imported. If standards of names for different types of texts could be agreed upon, a situation could actually be envisaged, where the same paper would be formatted very differently depending on which journal it would be sent to, and yet the formatting would happen fully automatically.

In addition to reading the named styles in certain word processors, Page Maker can also read style names placed in angular brackets, provided that the opening bracket is the first character to be met in a paragraph. This means that also ASCII import can lead to automatic formatting.

A further feature enhancing integration is that Page Maker keeps a watch on imported files to see whether they have been changed since they were imported, and if so it can perform an automatic update if desired. In cases where the authors are on the same network as the production unit this is a very important feature. It means that corrections to the manuscripts can be made locally by the authors, followed by an automatic update to the master-typesetting file.

This last feature made me perform some experiments. My preferred word processor is Word for Windows, which do have excellent desktop publishing features itself, although not to the same level as Page Maker. I had noted that its specifications for typesetting looked very familiar to those in Page Maker, and therefore I created two identical page set-ups, one in Word and one in PageMaker. The result was that paragraphs of text would go unchanged from Word to Page Maker, and when you made changes in Word you would find the similar changes in Page Maker. Well it was almost perfect. Occasionally you would find that hyphenation would change a little, but only occasionally. On the whole you could expect that corrections you made to the setting of the text in Word would be truly reflected in Page Maker. Thus it was possible to handle corrections after proof reading on a decentralised level, and having automatic update to the master Page Maker file.

The other point of integration to mention briefly is the use of standardised exchange formats.

The IBM DCA (Document Content architecture) was one attempt to create such a standardised exchange format, but it turned out to be too simple to support the formatting of complex desktop publishing documents. For instance it could not cope with different fonts. Today Microsoft's RTF (Rich Text Format) seems to be the de facto standard. We are using RTF quite frequently as it has turned out to provide a very reliable transfer, and then of course it is a 7 bit ASCII format, making safe transfer possible between systems, and by way of telecommunication. Unfortunately the support of it in word processors has so far been limited, but I believe that RTF or a comparable product eventually will become a standard and a must for any serious word processor.

There are, however, other elements in a publication for which a transfer standard is needed. There are vector graphics, and then there is the very special case of mathematical equations. Let me concentrate on the latter as it is quite interesting. One of the reasons for the popularity of TEX is its ability to produce mathematical equations. Mathematical equations are generally not a problem in archaeological typesetting, but of course in a CAA proceeding it is a problem.

We have chosen Math Type from Design Science to solve our problems with Mathematical equations, and that really seems to be a good choice. It is an extremely easy to use Windows based Equation builder. Apart from the ease and speed with which you can produce formulas, it has also the additional benefit of using postscript files for

saving. This means that you can import the saved files directly (with a bitmap header so you can see the equation), and then have them updated automatically, if Math Type changes them. The use of Post Script, which is a real page description standard, also provides us with another possibility. We can import the equations into a Word for Windows document, and due to the use of DDE we can actually activate the editing of an equation in Math Type by just double-clicking on it in the word document. When the document is finished it can be placed in a Page Maker document, and all the equations will be there. The only precondition is of course that all files must be in fixed positions within the same network.

The example of Math Type really shows the importance of using standardised file formats. By using Adobe Postscript the equations can just float around between the programs used in the production, and if a change is made to a equation, there will be an automatic update all around. This occurs through the DDE facility utilised in Word and by the automatic check on file updates carried out by Page Maker.

To conclude briefly.

Making a choice for a publication production strategy, it was important that we had a system that:

- * Provided us with acceptable professional typesetting capabilities.
- * Had a structured approach to typesetting.
- * Gave us the possibility of interactively intervening with the layout process.
- * Supported automatic formatting of imported text containing named paragraphs.
- * Supported automatic updating of changed, imported files.
- * Supported various standard exchange formats for easy and safe import and maintenance of data.

We found all this with Page Maker, and Page Maker it will be. But it could as well have been Ventura I am told. This, however, is of minor importance. What is important, if we look ahead, is that the professional group of Desk Top Systems seem to bring us a production platform, where increasingly, we will find all the features we need for good and efficient publication production within archaeology.