

About 80 people turned up at the Digital Publishing Forum's "Making an eBook" seminar on September 3 in Auckland.

Auckland branch secretary Adrian Blackburn, who attended, comments: Only a few NZSA members were among those present – and if anyone had hoped to discover the do-it-yourself secrets to eBook-making over a few morning hours, then they were doomed to disappointment.

Much of the presentation by G. A. Pindar and Son (NZ) managing director Gerard Reid and his tech specialists simply made it clear that a lot of specialist knowledge – and in Pindar's case some pretty clever programming -- is needed to make fully professional-looking eBooks using currently available computer software like ePub.

However, Gerard's clear outline of what is involved would certainly give authors who choose to go the eBook way a good idea of the questions to ask of their providers during the process.

It also made it very clear that if you are preparing an MS destined for eBook use, it's best if possible to have a completely uncluttered Microsoft Word document to provide to your specialist book producer.

Completely uncluttered means just that: a perfectly unformatted text document, one font, no different sizings, no bolds, underlines or italics, page numbers, heads, indents, paragraph separations.

Such simplicity, which is what the currently favoured ePub software requires, is beyond most traditionally trained typesetters, (they'll always throw in some formatting quite automatically and probably never see it as they check the copy) but the amount of time and trouble it takes if not done in this fashion can cost a lot of money. ePub is just not happy to swallow and digest anything else.

The message to me is that you prepare two copies of your MS, one uncluttered as above, the other with your preferred formatting which can be used as a guide a bit later in the process after ePub has made its initial contribution.

Photographs, graphs and tables present their own problems. It seems that they're best sized as they would be in the final publication and presented as PDFs, essentially as image files.

For the rest, it's worth taking a little time to read what an expert like Gerard has to say. Even the Luddites among us will probably have to engage with eBooks at some stage of our careers and a little understanding will help enormously.

Gerard's speech notes, lightly edited, follow:

eBooks
Gerard Reid
Managing Director, G A Pindar & Son (NZ) Ltd
Presented to the Digital Publishing Forum 3 Sep 2009

Welcome , and a bit of background on me and my company.

I have been in the book business for 35 years but in effect all my life, as my father was a Professor of English Literature and I grew up surrounded by books.

During the 1970s I was an Industrial researcher and taught myself programming in order to write statistical analysis programs to run on main-frames.

In the 1980s I ran the Publishers Association. I became aware of such advances as SGML which I promoted to publishers, unsuccessfully.

I also did formal studies in programming and systems analysis but, again, on mainframes.

In 1987 my wife, Mary Egan, founded NZ's 2nd digital typesetting business concentrating on trade directories. So our first efforts were in structured data.

In 1992 we produced our first database publication, "The Bateman Encyclopedia of NZ" and we have been pushing the boundaries ever since.

We started working in SGML publications in 1993 and, over the years, produced 37 dictionaries for Oxford University Press in Australia.

In 1999 we were the first local supplier to offer Computer to Plate technology and started designing workflows for publisher clients.

We produced our first eBook in 2000, a Rocket eBook format of “The Scarecrow” for Penguin.

In 2003 we produced our first full web delivered book, “Successful Email Marketing”, again for Penguin.

In 2007 we were bought by the Pindar group a Yorkshire-based print technology company, greatly strengthening our existing business foothold in the UK.

Our market base has drawn us ever more deeply into XML and eBook production and publisher workflow analysis.

So our work is practical in its application although we spend a great deal of time and money developing our own solutions to the increasingly technical demands of our customers.

eBooks Revolution of evolution?

There is a lot of excitement about eBooks with many new devices and operating systems coming on the market. I’m not going to talk about that aspect. Instead I’m going to inject some realism into the subject.

eBooks need to be understood in context or we will constantly be tripped up. We also need cool heads about the subject. Over-enthusiastic expectations have felled many in the history of book publishing.

A definition to start. The choice is not between books and eBooks. Ink on paper is a book, but so are digital images on screen. images on a screen. Either way you get a book. eBooks are simply another delivery mechanism.

Those of us old enough to remember, recall with sadness the hundreds of millions of dollars wasted around the world on CDs as a publishing mechanism. CD technology was invented in the early 1980s

By the end of that decade publishers put the two together and dreamed of offering books on CD. After enormous investment and high hopes it turned out that the public taste was not for that medium. Some kinds of content migrated (loose-leaf legal services and other regularly updated reference material) but, especially in the educational sector more fortunes were lost than made.

About 25 years ago I developed what I call a theory of Relative Displacement of Media. It says that the arrival of a new medium does not replace existing media. It only displaces them relative to each other.

For example the arrival of disc records in the late 1880s was predicted to spell the end of home entertaining, yet more homes have a musical instrument now than ever.

More recently we have seen the inaccurate prophecy of the death of radio and the cinema with the advent of TV, and the end of records caused by cassette tapes, CDs and Internet delivery. In every case the old medium has survived. I have recently ordered a family tombstone with engraving just like they did thousands of years ago.

So, no medium ever dies. And nor will the printed book.

The eBook, the printed book, the CD book, the audio book and other variants will co-exist. And each variant of the medium will be constantly re-invented like cinema with computer graphics or tombstones with digital etching. My prediction is that printed books will add value by incorporating ever-higher production values, 3D imaging, holograms, greater design emphasis, bar-code links, etc.

Which leads on to the question of why make an eBook version of a publication? But before that let's look at what we are talking about when we discuss books in their myriad forms.

If the form doesn't define the book what does? The content. Put in its crudest terms publishing is about acquiring content, doing stuff to it to make it accessible to an audience and then marketing it to that audience.

Implicit in that definition is all the things with which we are familiar: writing, illustrating, editing, designing, typesetting and page-building, indexing, manufacturing, marketing, fulfilling and so on.

But the purpose is always the same: to earn money by delivering content to end-users in a form they desire. Let's not lose focus — the end-user is best served when the content is delivered in the way they wish, when they wish.

So, we here, as creators, processors or deliverers of content need to think about this process.

What is this content at the heart of the task? It is almost exclusively the creative output of writers and illustrators or photographers. It is made up of units (words or images) that must be managed effectively to achieve the goal.

A whole industry has arisen around the concept of content management and, unless the book industry becomes part of it, it will pass books by. Thus the application of modern technology will be haphazard or ineffective in our sector. So, I'm going to ask for your patience while I cover the background to and evolution of content management that has spawned eBook technology.

eBooks and, before that, XML, have become topical only over the last few years but the background to them goes back much further. SGML (their antecedent) was created in the 1980s, long before the Internet. It proved to be the perfect starting point for solving a variety of Internet-specific issues. SGML stands for Standard Generalised Mark-up Language.

A working group completed most of the XML specification late in 1996 and it was formally published early in 1998. XML stands for eXtensible Mark-up Language. Most of XML comes from SGML unchanged. For example, the separation of logical and physical structures, the availability of grammar-based validation (DTDs), the separation of data and

metadata, mixed content, the separation of processing from representation and the default angle bracket <> syntax.

One change was that XML adopts Unicode as the document character set allowing such complexities as double-byte characters (e.g. Chinese, Arabic or Cyrillic languages).

However, for practical purposes only 95 of the total 128 characters of the ASCII set are usually employed in English or other Roman alphabet settings. Its 128 characters can be represented by 8 bits of binary code. And the unfamiliar 33 characters are machine instructions — even still including CR “Carriage Return”. achieved

The working group’s achieved goals were:

Internet usability.

General-purpose usability.

SGML compatibility.

Easy development of processing software.

Minimization of optional features.

Legibility.

Formality.

Conciseness, and

Ease of authoring.

Given that it is now 11 years old, it is surprising how slow the book industry has been to take up the opportunities XML affords.

What XML is:

An environment that allows a clear separation between content, structure and format. But, because we, in the publishing industry have, for so long, thought of these as inseparable, it can be difficult at first to grasp the distinction.

In terms of a book, a simplified description might say:

- content is what the author delivers to the publisher
- structure is what the editor does to it to make it comprehensible
- format is what the typesetter does to make it legible and appealing.

XML enables this separation through a mark up language (the ML of XML). In other words, by adding mark up, the content can be stored along with information that describes its structure.

The mark up follows a strict syntax defined by a specification called a DTD (Document Type Definition) or Schema, which, in turn, describes what the various components of the mark up do and how they relate to each other. It is not necessary to be able to read, write or interpret a DTD or Schema in order to be able to use XML — just as we don't need to know the details of what is under the bonnet in order to drive a car.

The three features of a Mark-up Language file (whether SGML, XML, HTML, etc) instantly recognisable to a layperson are:

- the various opening and closing tags, e.g. <head></head>
- the angle-brackets they are encased in
- the use of entity codes for unusual characters e.g. &oq; for opening quote.

The last of these enables an enormously large set of special characters to be used without having to have a computer or software capable of making sense of them. And this is what I was referring to when I mentioned the ASCII set. XML can be written entirely in ASCII and so work equally well on every computer.

What kind of file you are looking at will be clear from the declaration at the beginning

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd"><!mimetypeapplication/epub+zipPK      META-
INF/container.xml<?xml version="1.0"?>
```

What XML is not, despite the persistent reference to it as such, is a good database tool. While it meets the strict definition of a database, the technology of XML is about defining structure and being a stable medium of transport. The technology of true databases is about sorting, sifting and relating data.

XML is not a typesetting or formatting platform either. Yet some people try to force typesetting instructions (as opposed to structural definitions) into an XML file. The results are awkward, confusing and end up failing to achieve their goal. In passing, note

that XML works very well in highly-structured, regulated environments, e.g. as the output platform from a database prior to being input to an automated page-building process.

What XML does: It just sits there. Nothing at all itself. Which is probably why publishers took so long to see its potential. However, XML enables things to be done to a file that otherwise would not be possible. Just as the English language does nothing itself but enables things to be done with it.

The obvious question is why go to the bother, then? Because what can be done via XML adds power and value to the content. Having converted content into an XML file, a publisher can be sure that it is a stable legacy format. Because everything in the file has been reduced to the basic character set that has been the default for computers since 1963, a publisher can be confident of the longevity and transportability of the file.

Better still, XML is a robust base for multi-purposing. Whatever you want to do with a file, XML is a great place to start. From selecting parts of text based on some structural criteria to outputting to digital or print formats, an XML file ensures that the content is well-ordered, that editorial work doesn't have to be repeated and that methods of automatic or semi-automatic production can be utilised. This is why it is so important in discussions about eBooks.

Let me illustrate this by working backwards. (Shows image of a complicated typeset page).

Here is a page containing content, structure and formatting.

(Shows image of a page fully edited and structured but not made up.)

Strip out formatting and this is what you get. (Shows image of a completely unedited, unstructured text file.)

Strip out structure and what is left is content.

In all your future workings with books, I hope you will keep these in mind, so I will repeat them:

- content is what the author delivers to the publisher
- structure is what the editor does to it to make it comprehensible
- format is what the typesetter does to make it legible and appealing.

Each of these tasks has been relatively invisible to those who don't work in them. Authors & editors need know nothing about formatting, designers or typesetters need not know how to write. But from now on everyone is going to have to understand the difference in order to do their part accurately. For example an author who writes without a thought to structure is going to create a work that cannot, without great expense, be converted to an eBook form.

And this is where the hype departs from reality. Many companies (ourselves included) have spent large sums of money on trying to simplify or automate the production of eBooks.

Let's say you already have a file in Word. No matter how careful the author, there will be problems with it. For example, empty paragraphs (double returns) that are unstyled can trip up the production process. So, too, can irregular use of footnotes or end notes. Broken text blocks and many other little details get in the way.

One of our larger clients (also one of the world's largest publishers with nearly 5,000 titles a year) has had a team working on a simple tool to tidy up files. After several years they are almost defeated. All they want to achieve is imposing regular structure conforming to a DTD on the text. They have already decided that even with the tool, it is just too darned complicated and are discussing outsourcing the whole lot to us.

Next the structured text has to be converted to eBook format. We have just entered into an agreement with the developers of the world's leading eBook conversion software. Having secured many conversion contracts from the biggest publishers in the world, they found that more than 75% of the files supplied would not convert because they contained errors. Our role is to fix the files up when they crash.

The difficulty in dealing with that here is that it takes 15 years of development, years of training and the services of a full-time developer on staff to achieve the necessary performance. All I can really tell you is the kinds of problems that arise. And I'll cover these as I describe our conversion process.

For a variety of reasons, commercial publishers find it most logical to convert in two stages: first XML, then eBook. They want the multi-purposing and future-proofing potential of the XML files anyway. But also the process of XMLing tends to tidy many of the anticipated problems with then making the eBook.

eBooks

I have devoted so much time to talking about XML not only because it is usually, in a commercial environment, the preliminary stage before eBook manufacture, but also because it has allowed me to dig into issues of structure. XML, its parent SGML and other spin-offs such as HTML all depend on a rigorous analysis of content to understand its underlying structure and then give that expression within the tagging system.

The number one fallacy about making eBooks is that it can be done at the touch of a button. eBooks can be made that simply but only on a number of conditions:

- the book has an extremely simple structure
- the originating files are precisely made
- the structure is rigorously followed in the styling of the original file
- that there are no tables
- there are no images
- there is no or extremely simple frontmatter
- that links, if any, are correctly built

The single biggest difficulty in making eBooks is structural. 90+% of all problems stem from failure to have a structure or failure to define that structure or failure to express the structure accurately and intelligibly.

How we actually make an XML file and eBook

We start, as I have said, with XML.

The tools

We use a combination of off-the-shelf software and our own scripts. We are constantly revising the scripts to add power by automating more and more of the process.

The software we use of course includes Microsoft Word for the raw input and Adobe InDesign for some structuring, and then we mostly use BBedit which is really good for doing GREG searches and is a quick way of fixing most errors.

For those unfamiliar with GREG (global / regular expression / print) it is a command line utility that enables find & replace searches or, more formally, searching for lines matching a defined input string (a regular expression) and copying them to an output file.

Oxygen is used to check that the XML is validated, but you can open the ePub files in Oxygen and it will let you know if there are any errors in them. And then there are various utility programs such as Apple script , Adobe Photoshop, Adobe Digital Editions, Stuffit and epubcheck.

We start with a visual inspection. We check the file in its original (usually Word) format, e.g. make sure text is all styled, there are no missing links, images are grouped with their captions so they don't get lost in all the code, etc. If text isn't styled at both the paragraph and character level, it will just be exported to a default style, and important structural information will be lost.

We then export to a text file through a custom-developed filter to make rough XML. The file will be roughly styled at this point, but it won't be perfect, so we have a script that we use to clean up anomalies e.g. empty tags. The script also finds chapter markers, e.g. looks for chapter titles, and splits the chapters so they are separate files.

At this point we also create an .ENT file that lists all the entity codes I mentioned before, and a .MAS file that links all the files together and contains the imprint information. The files need to be separate but linked to help with the ePub part of the process.

At this point there will probably still be a little bit of manual clean-up to do, e.g. finding elements that aren't catered for in the script (that is where the structure has been customised by the file's originator in a way that doesn't conform to the DTD to which we are working), checking all of the images have been exported into the correct place, etc.

At this point, we validate the files against the schema or DTD, this can either be supplied by the client, or taken from the docbook default DTD, or you can tweak docbook to the client's standards. The validation tools are built into XML editing software.

Towards the end we use another of our own scripts to generate XML ids. These are unique file identifiers that permit the whole lot to inter-act as a single publication. They also allow the content to be identified at a highly granular level (separately identifying many different elements) enabling future re-packaging.

Then we package all the files together with the images. And the end result is a valid, tested and well-formed XML publication. (Shows image of an XML file.)

You may well be surprised at how many steps there are to get here. With XML what you are doing is taking a document and/or design and transforming it into a structure without any design elements. This is why automation doesn't completely work all the time, only the simplest of books can be automated, most books have varying elements that mean that there is almost always something to do manually.

Next we move to eBook conversion. But before I do I need to point out why I have consistently referred to eBooks up till now. eBook (or ebook or e-book) is the generic name for any form of digital file designed for reading on a hand-held device or computer. However, there are very many different kinds and each needs to be prepared separately. Despite Amazon's best endeavours, it seems that the ePub established as the official standard of the International Digital Publishing Forum (IDPF) in September 2007, and which is free because it is non-proprietary is in the lead at present.

But other standards have not yet died. These include the Kindle, Microsoft Reader, Adobe eBook Reader, Palm, Rocket, and about 12 others. Publishers have reasons (usually associated with rights management) why they select a particular platform and larger forces than common sense may well decide what becomes the norm over the next few years.

I mention this here because a separate production step has to be taken for each standard, though this is relatively straightforward. I will also only refer to making an ePub eBook because it is the most

common form we make. However, much of the process is identical except for the content of scripts we run.

In order to comply with the ePub standard, we need to meet 3 specifications: Open Publications Structure (OPS) 2.0 for formatting of content,
Open Packaging Format (OPF) 2.0 for the structure of the ePub in XML
OEBPS Container Format (OCF) 1.0 collects files as a zip archive

I'll refer to the latter two of these later as we make an ePub.

Remember I said, earlier, that we always make XML first because that sorts out many of the potential problems. I cannot tell you how to make an ePub otherwise because we just don't do it.

First we transform the .XML file, through an intermediate stage using Oxygen. This could be done manually if you want, by finding and changing the tags. However, this is laborious and error-prone.

We have not found any workable way to get from XML to eBook in one pass. So, we have another script that tidies up the intermediate file and places the relevant files into the appropriate eBook file structure. This script also creates the content.opf file I mentioned when talking about the ePub standard and the toc.ncx file, and references all of the images in the content.opf. These are the files that define the structure of the ePub. Again this can be done manually but takes time.

If the content and toc files don't match, you will get an error in your final ePub, this can be either references that don't link correctly, or just an entire missing chapter.

But wait ... we're not finished. In order to make the ePub presentable there can be quite a lot of tweaking involved, especially for non-fiction. Tables, images and boxes may all require manual work. The more complex a book is, the longer it will take. (one book, chockablock full of tables, boxes and equations for example, took us about 6 hours for this stage).

Images need to be converted to low res rgb jpegs, or they will not show on the reader.

So after all this has been done we create a zipped file of all the files relevant to the eP This is the OEBPS Container Format I referred to in the standard. But there is one crafty little tip: having zipped the file it is necessary to change the file extension from .zip to .epub. And you have made an ePub!

Now you need to check that the spine of the ePub is working. Spine doesn't mean the same in an eBook. It is analogous to a contents page and works slightly differently from the rest of the book. It can sometimes malfunction and macrons that work inside the book almost never do on the spine.

It is also a good idea to do a page turn on at least one device as you would for a print publication and check the file on more than one digital reader.

We also run the ePub through the google ePub check which is a free application that will find most errors (if there are any).

So what do the finished goods look like? (Shows three pages from a book as displayed on the Sony eReader).

It seems a shame. After coming so far, to end on a down note but there are a few other matters to consider.

No matter how complex this process might seem, once it is set up, it is really just routine. But there are a number of issues that must be watched out for.

The individual eBook text files have to be under 100kb to work on all readers. One of the main problems we have come across is that, while there is an ePub standard, not all companies manufacturing ebook readers take this into account e.g. the Kindle. What works on Stanza, may not work on the Sony eReader. So your eBooks need to be made to the lowest common denominator to ensure that they can be read on all of the available devices.

The devices are getting more robust, so the size of files they can process grows. We anticipate that the 100kb limit will soon no longer be a problem. However, eBooks will still need to be made to the 100kb limit so people with earlier readers aren't left out.

Fonts – there are a number of limitations with the font you can use in the ePub. First, if you package a font with the ePub it HAS to be a free open type font. EPubs can be unzipped and the fonts can be retrieved by the person who has purchased the ePub. Therefore, if you include a font in the package that is not free, you are in **breach of copyright**. We use Charis, as do most people, because it contains a lot of glyphs.

Second, macrons do not show on certain devices, this is especially a problem in NZ as macrons occur in a large number of our books, especially those in Maori. I anticipate this problem will eventually be solved as new updates come out for eReaders. Charis does include macrons in its glyphs. This is a tricky area and there isn't a lot of discussion on it on the internet, probably because not many other people in the world need to use them.

Third, not all readers use the font that has been packaged with the file, for example Stanza. Even if you package the font with the ePub, Stanza ignores this and the reader is able to choose from a list of available fonts on their iPhone or iPod.

Fourth, including the fonts increases the total size of the file by about 4Mb. This doesn't cause problems, but larger files mean that you can store fewer ePubs on your Reader.

Last, all our books are left-aligned in their design for the moment. This is because the use of hyphenation on different readers varies. Stanza hyphenates paragraphs wherever it feels it is needed (you can also change the alignment of the book in Stanza, e.g. to centred). Digital Editions and the Sony Reader (as far as I know), don't hyphenate at all, and it would be futile to put hyphens in ourselves because the ePub text reflows according to the size of the screen it is sitting on.

So a lot of the classic elements of typography can be lost in the ePub conversion, which some people may find distracting. So, when it comes to the overall look of a book, quite a lot of control is out of our hands.

My final comment is that the essential element for creating an ePub is a file that has been well-made in the first place. That is the place to concentrate attention to get best value for investment.