<title>BRIEFING PAPER</title>

<subtitle>XML for eBooks</subtitle>

Introduction

We are seeing today a mass move by many publishers and information owners to invest in XML for their content. Look at any publisher or any large, information-rich organisation such as the BBC and the Open University (but few other universities) and you will see evidence of XML uptake. It is more than a trend; it is accepted that this is an inevitable route to take to capture, manage and publish substantive volumes of information. The core underlying benefit in doing so is that this frees organisations from proprietary lock-in, and maximises content re-use.

It also provides a basis for longevity of investment – in an information sense, as well as a financial sense.

XML should not be an after-thought

XML is a W3C standard, derived from an ISO standard, so it has the characteristic of being safe. SGML proved difficult for many to work with, but has XML proved any better?

It has been demonstrated by some organisations that it is possible to "do XML" without impacting too much on established, traditional typesetting workflows. This, however, should be approached with caution, as actual use tends to be somewhat restricted. This is despite the fact that XML has been quite widely adopted, if only in a token sense.

For example, the most popular visual layout tools used commercially – tools such as Quark Xpress, Adobe InDesign, and even Microsoft Word – actually allow production teams to save their files in XML as a by-product of the layout process.

Earlier Briefing Papers have documented the difference between using XML 'properly' (e.g. in a true semantic fashion), and merely using XML as a "Save As ... " file format. This is not "doing XML", but merely acknowledging its use and potential late in the chain.

Many publishers now recognise the flaws inherent in this "XML as an afterthought" approach, for example:

> the application-specific file (e.g. Word, Quark) remains the master copy, and the XML merely an export, so in fact there is still the very real risk of proprietary lock-in;

However there is a much more serious issue with this. We wrote back in 2008 of the non-determinism of DTP or Word Process derived tools, i.e. that the same content may produce different layout on a different day from different typesetting engine. End-to-end XML-based batch production is capable of ensuring that hitting the release button again in 6 months, or 6 years, gives the same desired output.

You cannot rely on exported XML to be able to reproduce the precise layout

CAPDM Ltd.

of a document again, for the simple reason that the layout information is still embedded in the application files. With XML as an export, the only way to reliably re-use the XML is in the same version of the same application which exported it. This actually renders it pointless even as an interchange or archive format; it cannot be reliably re-used.

In fact the position is much more serious. The XML you get from exporting from a visual tool is essentially garbage. No visual layout tool (DTP, word processor) is going to be able to create re-usable, semantic, and well-structured mark-up – which is how XML should be used. The best you will get from an exported XML is a mapping of point-and-click styles to equivalent elements in the export.

XML should be semantic

The minute you want intelligent use of semantic data – such as author's names or copyright information to feed into databases, or for re-use in running footers - you find that exported XML is simply not good enough.

You will not be able to generate semantic mark-up of student quizzes – for example, allowing the answers to be laid out in an appendix in print but as a pop-up window in an on-line rendering. You will also not be able to embed metadata, such as learning objectives or keywords that enable alternative paths to be taken through the content and its on-line exploitation.

Exported XML will provide you with headings, paragraphs and character styles, and it will enable you to import it back into the exact same application you exported it from. But that's pretty much all.

This is not news – we, and many others, have been covering this for some time . This is precisely why there has been, for many years, a push towards pagination directly from XML where:

- there is no proprietary application format in the way;
- you edit only the XML;
- the layout of the final product is precisely reproducible from only the XML master a month, a year or a decade later.

Used to the full, the XML should be semantically rich enough to enable it to be used to create all of the varied print and on-line outputs needed from one source. That is there should be an adherence to the principle of single-source, multi-format (channel) publishing.

Nowadays cheap, accessible XML editors are readily available which can ensure that content is structured and re-usable.

---

[1] See, for example, gilbane.com

XML is quite unrestrictive

The established 'wisdom' is that batch typesetting direct from XML can only produce very simple layouts, and that it involves writing enormous, deeply unpleasant style sheets, buying extraordinarily expensive rendering engines, and retraining all of your production staff.

The established wisdom is not just out-dated, it is also quite wrong.

Professional batch typesetting from XML is established, it is highly effective, and it is capable of generating results as good as – and in many areas far better than – expensive proprietary DTP environments like Xpress and InDesign. It also has a radically different cost base, particularly when multi-format, multi-styled, and multi-lingual output is required. I will leave it as an exercise to the reader to decide which way this difference falls. You are welcome contact CAPDM if you want a typical spreadsheet showing the cost structures.

Conclusion

In a future Briefing Paper we will explore issues of professional batch typesetting in more detail. For example we will discuss:

- how it differs from traditional typesetting;
- the benefits and limitations;
- how external scripting is coded and applied;
- the level of typesetting control available with processing instructions;
- the process of specifying a batch style;
- complexities of tables and equations.

We will illustrate these issues with an example or two of wisdom derived from our 15 years of doing this for 20 or so education delivering clients. In this time we have seen a huge variety of structures, publication types, the rise of on-line delivery, and – hot off the e-press – the dawn of the eBook and Tablets.

We shall also explain pitfalls of some commonly adopted systems (for example, why XSLT is great, but how it struggles with the 'big' stuff), and we shall make some predictions about the future (for example, the potential of ConTeXt).

In the meantime, remember that...

> "there is XML, and there is XML"

Visit http://www.capdm.com/resources for more CAPDM briefing papers.

CAPDM Ltd.

CAPDM Ltd.

22 Forth Street
Edinburgh
EH1 3LH
United Kingdom

capdm.com
enquiries@capdm.com
+44 (0)131 477 8630
@capdmltd