WYSIWON'T —
The XML Authoring Myths

Tony Stevens

Turn-Key Systems

Abstract

The advantages of XML for increasing the value of content and lowering production costs are well understood. However, many projects fail to exploit the full value of XML because content is generated in some form that needs to be converted to XML. This process is always costly.

It is difficult to find technical or economic reasons to explain why XML authoring is not more commonplace. It is much more likely that any resistance can be explained by misconceptions and unreasonable expectations, mainly due to the widespread use of "What You See Is What You Get" tools. This paper attempts to challenge the prevailing culture of document authoring by presenting seven widely-held "myths".

Introduction

Since its official release in 1998, XML has become an integral part of the information processing landscape. Few would deny its ability to reduce production costs, and to increase the value of content through reuse and multi-purposing.

However, many publishing applications fail to gain the maximum benefit from XML because they incur unnecessary costs when creating XML content. It is not uncommon for authors working on the same project to deliver content in a number of proprietary formats, which must then go through an expensive and error-prone conversion to XML.

There is little doubt that authoring directly in XML is the most cost-effective approach when XML data is required. In the early days, it was understandable that people were reluctant to switch to new authoring applications that were unfamiliar and unproven. Now, however, these applications are much more mature and can provide an authoring experience similar to more traditional editors. There are many examples where XML authoring has been successfully implemented, and the predicted benefits obtained, yet using an XML editor is still seen as "hard", while using a word processor is "easy".

It seems that the barrier to widespread adoption of XML authoring is cultural rather than technical. This paper proposes some perceptions and prejudices that are probably

contributing to this situation. They are provocatively labelled "myths" as a challenge to those who have considered XML authoring and decided that it's not for them.

While this may seem like a condemnation of word processors and desk-top publishers, that is certainly not the intention. The What You See Is What You Get model of authoring is valuable for many applications, but (like anything else) causes problems when used inappropriately.

Myth 1 – You only need one type of document editor

Taking a broad definition of document, namely something that is written at a point in time so that it can be read later, there is a vast range of style and usage. For example:

- Inter-office memo
- Birthday party invitation
- Corporate annual report
- Australian legislation since federation

Given this diversity, it seems strange that anyone should propose we use a single type of tool for document authoring; yet this is a commonly held view. Thousands of years of experience with tools have shown that some degree of specialisation is a good idea. There's a reason no one makes a combination hammer and screwdriver — why should document authoring be any different?

To put this in perspective, consider a shopping list and a set of customer orders:

Fruit shop	Apples	$1.50
Market	Bread	$3.50
Market	Milk	$2.00
…	…	…

Acme Co.	X7-501	$2000
Ajax Corp	X6-001	$500
MyCo	X7-505	$2750
…	…	…

These are superficially similar, and we can do similar operations on them (sort them, group them, total and subtotal, etc.), but no one would suggest that a common tool should be used to create them. We use spreadsheets for shopping lists and relational databases for customer orders — although there is an overlap in their application areas, we understand the benefits of having separate tools.

One day, when the distinction between structured (e.g. XML) and unstructured (e.g. Word) documents is a given, we will wonder why anyone ever thought of using Word to create XML documents.

Myth 2 – Word doesn't cost me anything

Many people already have Word installed on their desktop. It's easy to think that this means the cost is zero. However, when you factor in:

- Development of templates, plug-ins and other customisations,
- Training authors how to use Word in the "right" way,

- Costs of manual intervention in the publishing process to handle documents that aren't "right";

the true costs start to become apparent.

For many people, the purchase price of new software and the cost of training authors is enough to prevent them from changing the way they author documents. This is fair enough, as long as you are balancing the one-off setup costs with the ongoing savings you would expect to make from having a reliable production process.

Myth 3 – One day there will be an automatic Word to XML converter

There are a number of tools that claim to convert Word documents to XML, or to assist authors in creating XML data. The fact that there are so many tools, each working in a different way, attests to the fact that there is no simple solution to this problem. After all, the potential market is huge, so if it could be done, someone would have found a way by now.

With an understanding of XML, it's not hard to see that it's theoretically impossible to create such a converter. An XML document can contain semantic information (e.g. a piece of text is a product code), whereas a Word document typically does not. An XML document is therefore more information-rich than a Word document, so an automatic converter would be creating information. To do this requires intelligence, and we have yet to create any artificial system that exhibits true intelligence.

It is sometimes claimed that semantic information can be included in a Word document by using paragraph and character styles. See the next myth for a discussion of this.

Myth 4 – If I use styles I can convert to XML

In theory, authors can be provided with an editing template that provides named styles to allow them to add semantic markup to their documents. In practice, this never achieves the goal of creating data that can be automatically processed, because no one ever manages to apply styles consistently.

The problem with styles is not due to author competence, knowledge or training; it is due to the very nature of modern word processors. The reason Word is the most popular means of creating documents is that it allows the author to do almost anything at any time. This is great when you are first setting up a publication, because you can get a result very quickly. It's not so great when you are in production mode and you don't want authors to be able to do things that will break the downstream processing.

Word has a very flexible development environment, and no doubt you could develop templates that would enforce the use of styles, and prevent authors from renaming or redefining styles. You might even be able to do it in a way that wouldn't break when the next version of Word came out. Whether this very costly development would be cheaper than switching to an XML authoring tool is another matter.

Myth 5 – One day Word will be able to create (useful) XML

Some time ago a new feature was added to Word – the ability to save a document in HTML format. While this was an extremely useful feature, it didn't threaten to make dedicated HTML editors extinct. The main reason was that the HTML created by Word contained lots of extra material in it so that it could be read again by Word. This extra material made the documents less suitable for use in a web browser.

The situation with XML is similar. Word has had the ability to export to XML for some time, but the tags inserted to contain the formatting information make it difficult to use. The most recent generation (WordML) is definitely an improvement, but still not a good match for most applications.

It's worth noting that creating XML data is trivial. I can take anything, put a start tag and end tag around it and call it well-formed XML. This fact should make us very cynical about anything that claims to "export XML".

This is not to say that an XML representation of a word processor document is useless – the fact that it can be fed into an XML processor opens up interesting possibilities. However, it's important to draw a distinction between "single-purpose" XML that contains presentation information, and "multi-purpose" XML that does not.

Myth 6 – WYSIWYG is the most efficient way to create documents

It's fair to say that What You See Is What You Get authoring has changed the way we think of documents and their creation. If you only want to print your documents, and only in a single format, it's difficult to see a cheaper and faster way of creating them than what Word offers.

However, if you do want multiple output formats, WYSIWYG is actually a liability. In theoretical terms, it has the potential to greatly increase production costs for other formats, because you have to remove one format and create another. It also increases the potential for error, because it makes it more difficult for authors to create material that is appropriate for all presentations.

Even for a single format, there comes a point where WYSIWYG costs more than it saves. For large documents, it can be much more efficient to automate the creation of formatting, rather than have authors do it manually. This is especially true of things like page breaks. There have been cases where authors spent a significant part of their time checking the page breaks in whole documents, even after very small changes. Removing formatting from the control of authors in situations like this can lead to substantial productivity gains.

Myth 7 – Authors can't cope with tags

This is probably a hangover from the bad old days of SGML. A lot of unnecessary complexity in SGML is due to its goal of assisting data entry (short tags, optional end tags, etc.). It can be argued that this actually made it harder to enter SGML data, because the author needed a great deal of knowledge about the way SGML works. This probably contributes to the perception that tags are "hard".

There is obviously no reason why a tag called "Heading1" should be harder to understand than a paragraph style called "Heading 1". In fact, it's usually easier to explain how start and end tags work than it is to explain how paragraph and character styles interact in Word (something I still don't understand completely).

Far from being an unnecessary complication, tags are an unambiguous way of revealing the metadata, which is just as much a part of the author's job as the text. Let's not try to hide them any more – let's make them work for us!

Conclusion

It's unlikely that setting up an XML authoring environment will ever be as easy as installing a word processor. The important question is whether this is a fault of the XML editors themselves.

In my opinion, making a good document is the hard part, irrespective of how you do it!

