1. WHY YOU SHOULD USE DITA FOR ALL CONTENT

As the silos of marcomm (marketing communication) and techcomm (technical communication) are growing closer together, companies are facing an important choice: which of the currently used content creation standards and systems are going to be used across these silos once they are truly integrated?

Replicate content, not chaos

As marketing has historically received much more attention – and budget – the obvious choice would seem to be making techcomm use the tools that are already in place for marcomm. In many cases, this will be Microsoft Word for all print-oriented content and WordPress for the web. This choice seems even more appealing when involving subject matter experts in drafting and reviewing content for techcomm: these experts are – as the name suggests – experts in their subject matter, not in the usual tools used for techcomm. Moving techcomm tools to the marcomm standard therefore seems to solve two problems at once. As everyone in the company is going to be using the same tools, content production should be easily integrated across the enterprise. Problem solved, or so it seems.

But things should rarely be taken at face value. In this case, the move toward Microsoft Word and WordPress (or similar tools), implies a retreat to technology that we have tried to get away from in the past decades. The reasons to move into structured authoring based on an XML standard are still very valid today. In fact, those same reasons apply to marcomm as well. This means that the reverse move – marcomm adopting the tools used in techcomm – is much more beneficial for an enterprise.

The reasons to move toward structured content creation tools are valid for techcomm and marcomm alike:

- Reuse without creating copies (which quickly become unmanageable)
- Automated tools to augment and improve content (imposing standard terminology and structure)
- Allowing product data to be integrated with content (again, without creating copies)
- Creating a wide variety of outputs from the same content using automated publishing tools

The huge investments that large enterprises have done in structured content technology were really defined by one single overall goal: creating better content while maximising its reuse potential. The move to XML as the basis for all content has been instrumental for this. In fact, most of the content creation tools in use in marcomm today (including Microsoft Word and WordPress) use some XML standard under the hood.

OK, but what about this DITA thing that techcomm is talking about?

Once the decision to use structured content is made, there is another – much tougher – decision to reach: which of the many structured content standards are we going to use? techcomm adopted DocBook, S1000D or DITA. marcomm often uses WordPress. Software developers may prefer Markdown. And these are just the mainstream standards – there are many smaller or entirely proprietary ones. Using semantic tags – as opposed to the format-based tags of HTML or the much older RTF – means that you are wrapping pieces of content into meaningful markers: a sentence might be a procedure step, a note, a supply, a tool, or anything else that has meaning to the structure of your content. These meanings allow defining templates for particular types of content: a maintenance task starts with a requirements section that lists supplies, tools and warnings, followed by procedure steps.

So which semantic tags are required to structure your content? Well, this depends on the content and the business domain for which it is being created. This is the reason for the incredible pace of adoption for XML, or eXtensible Markup Language. In less than 2 decades since XML was defined, literally 100s of XML standards have been published. Each business domain has defined their own standard set of meaningful tags to be used. Everyone in the aviation industry uses the same set of S1000D tags, so that everyone knows exactly what everybody else in that domain is talking about. Other domains have defined very detailed semantics to describe semiconductors, medical devices, cookbook recipes etc. Wherever the need for more specific semantic markup was felt, a new XML standard was created.

The defining difference between DITA and all of the other XML-based standards is exactly this: DITA is truly XML whereas all the other XML standards are merely XML-based. What I mean with this is that you can no longer add your own set of semantic tags to an XML-based standard without breaking the rules of that standard. The only choice you have in extending DocBook is to join the standards committee, convince the other of the merits of your proposed extensions, and then get the tool vendors to implement the support for your extensions. Finally, after what in real life will be a period of 5-10 years, you may be able to start using the new tags you so dearly wanted to improve your content.

In DITA, you can define new semantics and start using it immediately, without the need to convince the standards committee or tool vendors. As long as you play by a small set of so-called specialisation rules, you are fine and your content will remain true to the DITA standard, even with the extra tags. Tools – for editing as well as producing various output formats – do not need to know anything about these extra tags except their ancestry: which tag is the new kid on the block based on. Whenever a tool encounters such a new tag, it will simply treat it as its ancestor. Only if you want to do something special with the new tags in your tools do you need to extend the tool's capabilities. But for most purposes, just having the ability to define more detailed and fitting semantics for your partiular business domain is sufficient to improve the semantic quality of your structured content.

Does every author need to become a techie, then?

Does every driver need to be a car mechanic? As stated before, even Microsoft Word is using XML. This shows that what is visible on the surface does not need to look at all like what is under the hood. Allowing non-technical authors to create XML-based content efficiently is a matter of choosing the right tools that hide all that complexity. The pioneering days of DITA, when authors were exposed to all the tags and angular brackets, are long gone. Modern tools for structured authoring allow easy restyling of content, much in the same way that output can be restyled for various devices and formats.

The key in making DITA usable by non-technical authors is in hiding the complexity under the hood. This entails a couple of things, which are easily managed in DITA and not so easy in other XML-based standards:

- Restrict the available semantic tags to those which are meaningful and required in your enterprise
- Hide some elements and attributes (allowing the tools to maintain them as required)
- Add tags where needed to make the content fit your business domain better
- Allow various editing tools to access the same set of reusable topics
- Allow conditionalising content to increase the reuse potential of topics across techcomm and marcomm
- Use a central repository of reusable content snippets (such as warnings and notes)
- Have a centralised set of tools for the production of all output in all required formats
- Allow easy interchange of content (source materials) with other companies (suppliers or B2B customers)

Conclusion

Choosing to not use structured authoring methods for all content in a modern enterprise is simply not an option. This is proven by the incredible success of XML as a basic technology. The next step in optimising the production of content lies in making these structure authoring methods available across the enterprise. This will enhance the quality of marcomm and allow reuse of content between techcomm and marcomm. The only XML standard that can live up to this challenge is DITA.

VTOPIAE INSVLAE FIGVRA
Ciuitas Amaurotum
Fons anydri
Ostium anydri