THE BEGINNER'S GUIDE TO SMART CONTENT

TABLE OF CONTENTS

01 Background - How We Got Here

04 What is Smart Content?

06 A Brief Primer on Publishing Processes

11 The Cost of Smart Content

13 Who Else Is Using XML for Document Production?

14 What's Wrong with XML?

17 Smart Content Details

21 12 Reasons to Adopt Smart Content

BACKGROUND - HOW WE GOT HERE

In 2008, Quark launched a new initiative: Enterprise Dynamic Publishing. The goal was an end-to-end, automated publishing solution - create, manage, publish, deliver - for high-value communications. "High-value communications" is Quark's shorthand for content you sell, content that helps you sell, or content that helps you run your business. The company aimed to achieve this by enabling non-technical authors to create structured, semantically rich, XML-based content and then publishing this to various output channels.

Many of Quark's team members had years of experience in developing, selling, and implementing native XML authoring and publishing software for technical documentation, and we all recognized that if we could bring that process to non-technical documents - to customer facing, richly styled and branded communications - our customers could realize the value that tech-pubs department had been experiencing for tens of years:

- Lower cost and effort to create and reuse content
- Dramatically reduced translation costs
- Shortened time to market
- Higher quality content that is more accurate, consistent, and up-to-date
- Available in multiple formats

BACKGROUND - HOW WE GOT HERE

In order to apply this process to high-value content, there were some significant challenges to address. The two biggest challenges were:

1. Most native XML authoring tools were "XML in-your-face." That is, they presented a very technical user interface that highlighted the XML. And while every XML authoring tool attempts to "hide the XML," most users of these authoring tools would learn to work with the "tags on." These tools were not going to be accepted by the majority of non-technical authors such as product marketers, copywriters, financial analysts, legal analysts, and other knowledge workers and subject matter experts.

2. Most XML-based publishing engines were pretty good at formatting technical documents, but very few were capable of doing richer designs. Nor could they provide features in support of what marketing departments would consider a minimum baseline of formatting requirements such as support for CMYK printing, spot colors, text wrapping, complex shapes and transparency. And nearly all of them required programmer-level skills to successfully define the "stylesheets" required to make them work.

By the way, neither of these two statements has fundamentally changed in the intervening years. Most XML authoring tools are still too technical for business users to adopt and almost every XML-based publishing engine is still not appropriate for multi-channel, richly-designed layouts and professional output.

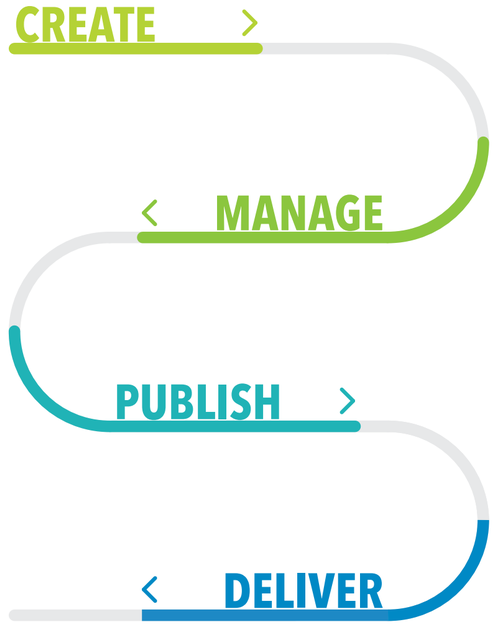
BACKGROUND - HOW WE GOT HERE

Quark had one of the world's most powerful and capable formatting engines in QuarkXPress and QuarkXPress Server which had been used for manual and automated publishing of a wide variety of high-value content including magazines, newspapers, marketing collateral, and product catalogs.

So it became clear that if we could leverage the QuarkXPress design capabilities, QuarkXPress Server's highly scalable automation, and integrate a user friendly XML authoring tool then we could help thousands of companies improve their high-value communications process and results. What's more, Quark's powerful Publishing Platform would be the glue to bring all of the parts together by providing user collaboration, component content management, and a rich API for integration with existing business systems.

For the past eight years we have been working continuously toward this goal. Like any trail-blazing path, it has not been without its difficulties and missteps. But today, with the help and partnership of numerous customers, we're proud to offer the only end-to-end system for creating, managing, and automating the multichannel publishing of high-value communications. Leading companies around the world now rely on this technology to do what a decade ago was unthinkable. There has been strong success in several industries including financial services, manufacturing, and government.

Which leads us to today and smart content...



WHAT IS SMART CONTENT?

Since the nineties, bold promises have been made about XML and what it could make possible in the realm of content reuse and automated publishing. In some areas such as tech pubs this has happened, but when it comes to mainstream content creation by business users and knowledge workers, XML is no more prevalent than it was 20 years ago.

Hopefully the term Smart Content causes you to think, "Something is different here. I know what content is, but what is Smart Content?" That's our aim because it's time to think differently about how we create content.

At a 30,000 foot-view, Smart Content is Quark's name for the next generation of XML-driven authoring and automated publishing of high-value communications. It reflects the fact that, while XML may be critical as an underlying technology, XML must to be relegated to the background when it comes to content creation and reuse.

WHAT IS SMART CONTENT?

At a very granular level, Smart Content is Quark's open, customer-configurable XML-based content schema. Quark's enterprise products are focused on understanding and implementing Smart Content so that customers see a return on the investment of adopting dynamic publishing as quickly as possible.

smart con·tent

[sma-hrt-kon-tent]

Open, customer-configurable XML-based content schema

Since the core of Smart Content is an XML schema definition, it's almost impossible to describe Smart Content without getting into the technical details. Similar to the hugely popular but seldom understood term "Big Data," the underpinnings of Smart Content are technical but understanding its value doesn't require a computer science degree. However, it helps to understand the basics of why XML - and specifically "semantically-rich XML" - is so powerful.

A BRIEF PRIMER ON PUBLISHING PROCESESS

Traditional word processing tools - what most non-technical authors use - such as Microsoft Word and Google Docs, allow an author to create any content in any order with any formatting the author chooses.

So an author can create a sentence like:

"**Quark** has released **Quark Publishing Platform** 10.5 that offers support for **Apple**'s new OSX version *Yosemite*."

Clearly the author is trying to communicate more than just the words themselves. The use of bold for the company names and green for the product release and italic for "Yosemite" is trying to communicate something, but you, the reader, may not know for sure what that extra formatting is trying to convey. If this was in a Web document, you might assume that the highlighted words are links that would take you to other documents, or maybe display a pop-up that presents information about the word. However as a source word processing document, the characters just have a different presentation.

What happens next in the content production process? In the most common corporate publishing processes, the content is created in MS Word then sent to an agency, the marketing department, or an in-house publications department. There, a "designer" will copy and paste the content into a desktop publishing tool such as QuarkXPress or Adobe InDesign and re-format the document for print or PDF.

What about all that formatting that the author did? Well, hopefully the designer understands what the author meant… but often this is not formally communicated nor is it in any way controlled, so one author might use colors and highlighting to mean something different from another author using the same colors and highlighting. The author may also be completely unaware of the design guidelines that the designer has to work with and what's possible as far as formatting is concerned. So the designer has to communicate directly with the author, expending time and energy for both, to understand the meaning of the extra formatting and then map this to the available styles.

A BRIEF PRIMER ON PUBLISHING PROCESESS

In this scenario we can call the designer the "design bottleneck" because they can't help but slow down the process as they have no choice but to use manual labor to create the final product. Many companies, especially within marketing departments, have switched to a web-first production process, but the problems, redundancy and manual labor is the same - just in a different order.

Also common in corporate publishing, once the first output design is done, a copy of that output is sent to the next stove-pipe process. In the case of print-first, the content is then sent to a Web designer (Web bottleneck) to copy and paste into an HTML document that is then formatted. And if those highlights and formatting characteristics in the original Word document are meant to be links and pop-ups, then more time is spent communicating with the author and more manual labor is spent to enable those very simple, interactive features - including locating any interactive media and graphics that need to accompany them.

A BRIEF PRIMER ON PUBLISHING PROCESESS

Add in review cycles, long work queues for the graphic designer and web designer, the errors that are generated through manual cut and paste between tools from different vendors, and ultimately hours days, or even weeks can be added to the production cycle of even the smallest of communications. You may ask does this really happen and the sad answer is yes. We see it at some of the largest companies in the world that may have state of the art production facilities, killer products, drop dead apps and Web sites and yet behind it all are workflows just like this one.

If that original sentence was written in semantically-rich XML, it might look like this when stored:

<paragraph type="announcement"><company status="private" hlink="www.quark.com">Quark</company> has released <product hlink="www.quark.com/Products">Quark Publishing Platform 10.5</product> that offers support for <company status="public" ticker="APPL">Apple's</company> new OS version <product hlink="www.apple.com/OSX">Yosemite </product> </paragraph>

Which is admittedly difficult to read with all the markup showing. If an author had to manually write all of those XML tags… or even see those tags while authoring, they would likely revolt against the IT or business-unit leader that forced them to write this way. XML might be good for the business but it's not good for the author in this instance.

A BRIEF PRIMER ON PUBLISHING PROCESESS

However, lets look at what those tags and attributes are conveying:

<paragraph>...</paragraph> is a start and end "tag" which tells the formatting tool to add a line above and below the sentence, very similar to what every word processing tool does under the covers.

type="announcement" is an "attribute" of <paragraph> and it qualifies or "specializes" the paragraph. This could be used in many different ways through automation. For example, it might cause special formatting to be applied to the whole sentence. Or, it could let a search engine return better results when someone does a search for "Quark" and "Announcements"

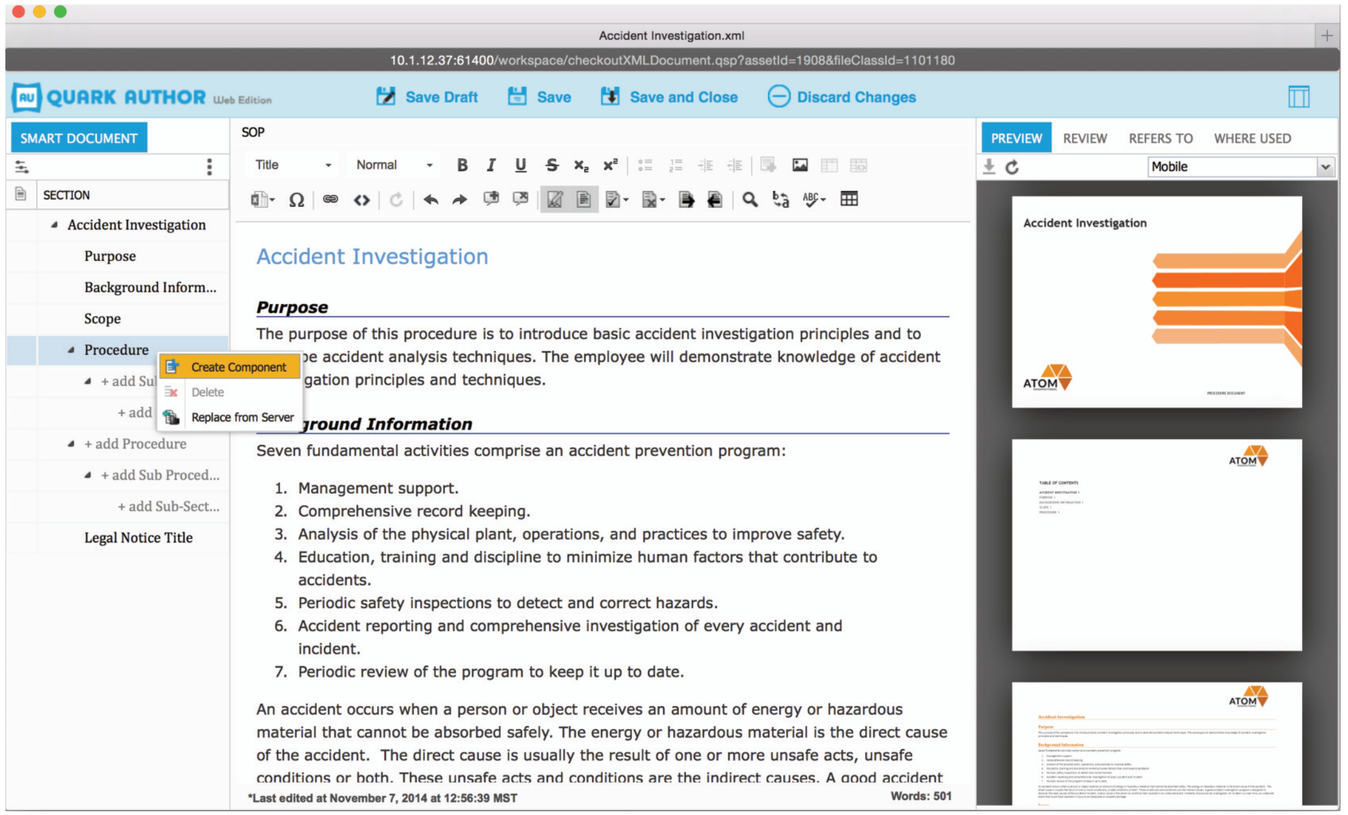
<company>...</company> used for both Quark and Apple similarly could - through automation - trigger unique formatting or even better, automate the process of embedding the link information as well as provide a pop-up on a web page when a user hovers over one of these names.

ticker="APPL" attribute on company would enable the web system to provide stock data automatically if a user hovers or clicks on "Apple's."

And so on.

What we just described is how XML is applied in Smart Content. It allows the author to create semantically meaningful, metadata-rich content that captures not just the words to be read, but the meaning and purpose of those words.

A BRIEF PRIMER ON PUBLISHING PROCESESS



In a user-friendly XML authoring tool, like Quark Author, the content creator doesn't see the XML, nor would they have to work any harder or longer - though maybe a little differently - than they did in a word processing application. The authoring tool would make the addition of those tags and attributes as easy as applying character formatting, but the resulting content is significantly improved. Now, instead of arbitrary and meaningless formatting, there is semantically-rich detail that any human or importantly, an automated system can understand and take action upon. And Smart Content using XML has many other benefits, not the least of which is that the source content - the XML - is in a non-proprietary, simple ASCII format for which there are hundreds of open source and commercial products that can process it. That means your content is yours, and you are not locked into a single vendor's tools to use your content. That's huge and what's more, your content is in a format that is ready for whatever media channels come next!

THE COST OF SMART CONTENT

"That sounds great" you might think, "what's the catch?"

Unlike a free-form word processing tool, if you're going to create Smart Content, it requires your organization to define the structure of all of your content upfront before a single word is written. Specifically the organization - often with the help of professional services experts - will have to define the semantics of how XML markup will be used.

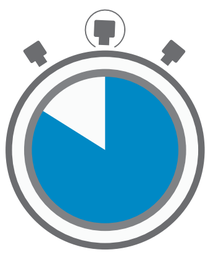
Not unlike creating a new spreadsheet, where you have to define the rows and columns, the calculations, and the presentation (formatting, page breaks, etc.), configuring an XML-based system requires planning, analysis of the content, its purpose, and the goals for presentation, interactivity, and output formats.

Once the semantics are defined, the publishing tool has to be configured for how to present those semantics in all of the various desired output channels.

And these two steps, along with configuration of content management processes including storage rules, workflows, and version management, can take several months to complete.

THE COST OF SMART CONTENT

But when Smart Content is appropriate, the results - the return on investment - is extremely valuable - productivity goes up, time to market is reduced, your company can support more information products without adding resources, and the end results are of higher quality. What's more, if you work with a company like Quark that has been doing this for many years, we can guide you to maximize your success.



So Smart Content isn't for every document nor for every purpose. It is best deployed when the content type has characteristics that include one or more of the following:

- High volume of similar documents
- High volume of revisions
- Frequently repeated creation processes
- Government or corporate regulated documents
- High possibility to reuse content across multiple documents
- Integration of data into the content
- Translated to multiple languages
- Delivered in multiple formats
- Delivered with multiple different presentation styles

WHO ELSE IS USING XML FOR DOCUMENT PRODUC-

There are many XML schema for authoring and publishing in the marketplace. Some are very generic and some are industry specific. Interestingly, even HTML4 and later versions are actually implementations of an XML schema called "XHTML." Other popular XML schemas include:

DITA
One of the most popular XML schema for technical document authoring and publishing. It was originally developed at IBM and moved to OASIS as an industry standard for technical publications. More on DITA later.

Docbook
A precursor to DITA and used heavily in technical publications and reference books.

MSP
Used in the United States, Australia, and other partner countries for capturing and sharing intelligence research at the Department of Homeland Security and across nearly every department of the US government. Supported for authoring through Quark Pubs-XML Accelerator.

SPL - Structured Product Labeling
Used in the United States for submitting drug labeling information to the FDA for approval prior to releasing a new drug or packaging to market. Supported for authoring through the SPL Accelerator for Quark XML Author.

And there are many more, including some companies that define their own custom schema from scratch which is A LOT of work, difficult, and expensive to do well.

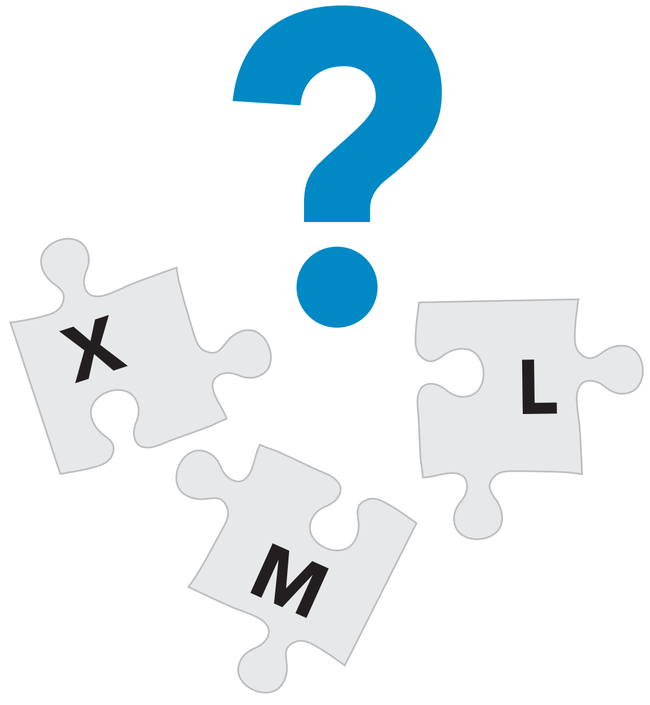
If there are many XML document schema available, why did Quark create a new Smart Content schema?

WHAT'S WRONG WITH XML?

XML for document production was first adopted by the technical publications industry. It is heavily used in Computer Software and Hardware documentation, complex discreet manufacturing, and some process manufacturing where the content is ultimately published as print and PDF, HTML, and several Help system formats such as HTMLHelp, MSHelp, EclipseHelp, WebHelp, as well as other output types. The most widely used document XML schema were created by and for the technical publications industry including the very popular DITA schema.

The result is that these schemas are extremely powerful tools, but are also extremely complex. To steal a quote from a Quark professional services partner, "DITA is great if your authors can think like programmers." That's perfect for technical authors who are, by nature of their jobs, highly technical and well trained. They are also full-time authors.

But for high-value communications, for example documents written by financial and legal analysts or product marketing teams, it is unreasonable to think that these part-time authors can or want to "think like programmers."



WHAT'S WRONG WITH XML?

What makes these authoring schema hard? They are often overly restrictive. At Quark many of our early adopters that used one of these schema complained that the simple task of cutting and pasting content from one area of a document to another area of a document was blocked by the application. Why was it blocked? Take the following simple example of a title and a paragraph (we're showing the XML tags, but remember that most XML authoring tools try to hide the tags):

<title>How to Make</title>
<para>Begin with the ingredients from the <keyword>Thanksgiving Recipe </keyword>.</para>

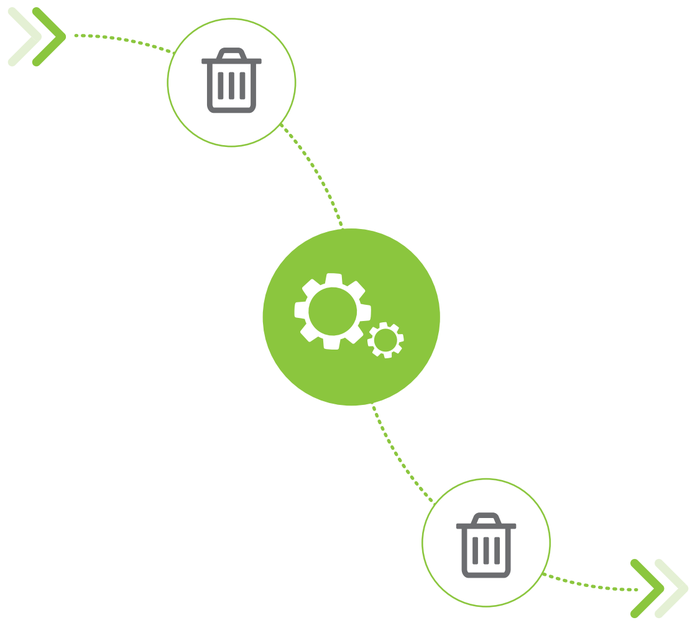
If the user selects and copies the phrase 'the <keyword>Thanksgiving Recipe</keyword>.' and pastes that after Make in the <title> then the authoring tool might block that paste because the controlling schema doesn't allow <keyword> inside a <title> element. That's frustrating, and worse, the reason for the failed paste is often hidden from the user - they can't figure out why it's blocked so they think the tool is broken. Of course a trained, full-time technical author would have a good idea what happened, would turn on "show tags" in their tool of choice, and only select the text they wanted - skipping the keyword tag. This is a simple example but many, similar use cases exist. It's a problem the Quark team refers to as "gross-edits," and is a significant issue when it blocks a business user from authoring with the ease they are used to.

This example highlights one of the major challenges for any XML authoring tool vendor, and especially for Quark who is targeting non-technical authors. The challenge is one of trying to impose rules and restrictions on users that have years of experience using free-form tools. Additionally, creating a user experience that manages and exposes those rules and restrictions to the user - without making the tool overly complex - is extremely difficult. That's why the user experience of so many XML authoring software products is more similar to a programmer's interactive development environment than it is to a word processing tool.

WHAT'S WRONG WITH XML?

Remember, the reason this challenge is worth tackling is the value of applying automation to the high-value, multi-channel communications process. Generally, the automation value proposition is relatively simple to describe:

- Automation lowers costs, improves quality, and shortens time-to-market.
- For automation to succeed it requires that the inputs are valid and expected: "Garbage in, Garbage out," as the saying goes.



So for Publishing Automation to succeed, the input - which is the authored, narrative content - must be expected and validated. That's where XML is powerful, because it is easy to validate and forces authors to only create what is expected. But it is also where XML authoring tools cause the most problems, because they only allow what is expected and content which can be validated.

Business users, part-time authors, and subject matter experts who have used a free-form word processing tool their entire career such as Microsoft Word or Google Docs, have expectations about how fast they can write and how much freedom (often total freedom) they have in how they write their document. Switching these types of authors to a controlled, "structured" content authoring tool that limits what they can do presents a significant challenge to the authors. The more prescriptive and restrictive the XML schema is, the bigger the gap between the author's expectations and their experience with authoring XML. And resolving that challenge is what led Quark to develop the Smart Content schema.

SMART CONTENT DETAILS

For the XML savvy, the Smart Content schema borrows ideas from many other XML implementations including, importantly, the idea of content types – sometimes called content classes or information architectural forms. The core idea is relatively simple: there are a set of fundamental types of content and all other content can be described as belonging to one of these root classes. For those familiar with DITA, another way to describe this would be "specialization" of one of those root classes. The concept of root classes and class hierarchies is common in computer programming, biology, physics, mathematics and more.

The value of root classes and class hierarchies is that a system that knows how to process the root element can provide basic processing of any specialization of that root without previously knowing anything about the specific specialization.

This is less complicated than you might think. By a simple example, if the system knows that all <para> elements should be presented with a blank line above and a blank line below, then if the system processes content that includes <para type="blockquote"> it will at least get right that a Block Quote should have a blank line above and below. There are many other processing rules, presentation rules, and user interactions that can be applied to all content of similar types. The "specialization" is created because a system could also add new and unique processing such as right and left indents for presenting a Block Quote.

SMART CONTENT DETAILS

What are some of these root classes? Smart Content represents these in different categories, and here is a table that compares some of the terminology that Smart Content, DITA and HTML use:

CONTENT TYPE	SMART CONTENT	HTML	DITA
Sections	sections	div	topic
Blocks	p	p	p
In-lines	tag	b, i, u, etc.	phrase
Lists	ul, ol	ol, ul	list type="type"
Tables	tables	tables	tables
Images	image	img	image
Media	Media	video, object	object
Metadata	XML meta fragment	tag attribute = "value"	tag attribute = "value"

SMART CONTENT DETAILS

How specialization of these root content types is handled in each markup language is one of the important differences:

In HTML, specialization of a root HTML tag is usually done to drive the CSS formatting or to trigger tag specific javascript and is most often encoded using a 'class' attribute such as:

…

However, in HTML, there are very few rules about where and how you can use and there are no rules on the value of the "class" attribute, so HTML is actually very freeform and not useful for high-value communications content authoring – though it is great for presentation in a web page or mobile application.

In DITA, specialization of a root DITA element such as <topic> is encoded like this:

<concept class="- topic/topic concept/concept">…</concept>

It's beyond the scope of this document to explain why the class attribute has such an apparently redundant value, but it's easy to identify the goal, which is that the element "concept" is of the class "topic" and therefore should be treated as a topic except where specific processing for concept has been defined.

In Smart Content, specialization is encoded like this:

<section type="purpose">

This is very similar to the HTML method for specialization, but has very specific implementation rules so that, for example, authoring a Standard Operating Procedure document can limit each document to one and only one "Purpose" and that Purpose must be after the title of the document. HTML doesn't limit the use of or even validate the value of class attributes.

It's worth highlighting that in HTML and Smart Content, the element name is always the root of the class. It is:

<section type="mySection"> it is not <mySection class="section">.

DITA users and other XML experts might ask, "Why not use the DITA method for defining specializations?"

SMART CONTENT DETAILS

The full answer is complex, but the simple answer is directly related to the difficulties described earlier in providing good authoring usability including support for gross-edits by cut and paste across one or more documents.

Nearly all-available XML parsing tools validate the structure of a document based on the element name (valid structure means that all the elements used are allowed by the schema and are in a valid order). Also XML parsers ignore attribute values when validating structure. By using the HTML style of element specialization, Smart Content can enable gross-edits with a positive user experience. The user can cut and paste an element and after the paste, added processing can either silently correct the type attribute, or if there is more than one choice that could be made, provide the author with a user experience that allows them to make a valid type choice.

While there are many other reasons for how the Smart Content schema is architected, this ability to "fallback" to processing based on the root class is one of the biggest and most valuable.

12 REASONS TO ADOPT SMART CONTENT

Even though the Smart Content Schema is relatively new in XML schema terms, its development has been grounded in years of XML, content authoring and publishing expertise by Quark and our customers and partners.

The schema is being used by a number of customers in different industries with great success. We welcome feedback on the schema and plan in the future to make the specifications widely available for other companies to use.

This is the age of Smart Content - the next generation of content authoring and publishing that is essential to support the always on, multi-channel, digital world that we need to communicate within.

7 Business Reasons to Adopt Smart Content:

1. Lower cost and effort to create and reuse content
2. Makes it easy for business users to create XML with out being exposed to it
3. Faster time to market
4. Reduced translation costs
5. Higher quality content that is more accurate, consistent and up-to-date
6. Enables automated publishing of high-value customer communications
7. Built to support multi-channel, highly-designed, interactive content that meets your brand guidelines

5 Technical Reasons to Adopt Smart Content:

1. Support for a superior authoring experience that balances the need for structure with ease of use.
2. Ability to fallback to processing based on root classes
3. Easier to configure and maintain than other schema such as DITA
4. Doesn't require complex programming to obtain the output formatting and design required
5. Lower barriers to adoption for non-technical users

ABOUT QUARK SOFTWARE INC.

Quark's software enables organizations of all sizes to meet customer demand for engaging, relevant communications when, where, and how they want them. Our solutions combine the power of XML with flexible layout and design to automate the delivery of customer communications to print, Web, and interactive experiences on the latest digital devices. Financial services firms, manufacturers, and governments around the world rely on Quark solutions to elevate customer communications to new levels, reduce time to market, and lower costs.

To find out more about Smart Content and how you can apply it to your business, speak with one of our experts by visiting www.quark.com/contact

For further information on Quark Enterprise Solutions, visit www.quark.com.

©2018 Quark Software Inc. All rights reserved. Unauthorized use and/or reproduction are violations of applicable laws.
Quark and the Quark logo are trademarks or registered trademarks of Quark Software Inc. and its affiliates in the U.S. and/or other countries. All other marks are the property of their respective owners.