Separation of Content and Form

Abstract
In the field of technical communication, there is a migration (driven by the need for efficiency) from style-based, document-centric writing approaches to topic-oriented, modular, structured authoring techniques.

Fundamental to structured authoring is the concept of the separation of content from presentation and delivery. The way a piece of text looks during authoring is irrelevant. The formatting and presentation are post-authoring considerations, and activities possibly not even performed by a technical writer.

The prime benefit of writing modular content suitable for publication in different forms for different purposes in different contexts is efficiency: being able to create more in less time. This benefit is not only applicable to the field of technical communication, but also in other areas of written communication practice where the drive for greater efficiency suggests a re-evaluation of writing processes.

Technical Writing Approaches
Technical documentation is a broad term which encompasses computer software and hardware user guides, corporate policy and procedure manuals, Help systems, scientific and medical publications, engineering manuals, machinery instructions, reference materials, and many other forms of non-fiction, corporate, product and business discourse. The word “technical” in the term is sometimes misleading, because the term also encompasses some forms of organizational communication.

Technical documentation is commonly produced by professional technical writers. For many years, technical writers have followed a document-centric, linear, narrative writing paradigm, treating a manual as a self-contained and isolated work1. Before computerisation, technical writers wrote drafts in longhand before sending them for typing, rewriting, editing, reviewing and typesetting. When word processing software tools were adopted by technical writers, document-centric authoring programs such as WordPerfect, Word and FrameMaker allowed the same style-based, document-centric paradigm to be used, but with the technical writer taking over the former roles of typists, typesetters, layout artists, and in some cases, printers2.

When topic-based, modular writing techniques became known in the early 1990s, the concept of single sourcing became practicable3. Single sourcing is "a method for developing reusable information"4, where reusable document modules are assembled to form publications, with different combinations of modules resulting in separate publications. Modular writing is a technique that makes single-sourcing possible. Modular, or topic-based, writing is a style of document design and architecture where content is structured into independent, small modules (topics) which can be assembled into one or many larger texts, such as a books, Web sites and Help systems. The advent of documentation technologies based on eXtensible Mark-up Language (XML), at the start of 21st century, expanded the possibilities for single-sourcing and reuse, and led to the adoption of the philosophy of separation of content and form through semantic mark-up5. XML is a set of standards for the categorisation, storage and retrieval of all forms of structured information. Practically speaking, XML is a set of rules for creating information structures, which are known as "XML applications".

References
3 Robidoux, (2008). p 111
In technical communication, the dominant XML application is the Darwin Information Typing Architecture (DITA). DITA is a semantic mark-up language which incorporates the ideas of topic-based, modular architecture, standard information structures, and the separation of content and form.

The following schematic shows how this separation impacts a technical communication team.

DITA has no native presentation format. Mark-up or tagging within DITA simply labels what the information is, and not what it looks like. A phrase within a sentence may be a window name, and that is how it is marked up. This approach is known as semantic mark-up. Whether a semantically marked-up phrase eventually gets displayed in bold, or in red, or in a box, or not at all, is determined later, outside the authoring process.

Documents are never displayed to the reader in DITA; DITA is not a presentation format. Content is almost completely separated from presentational form and delivery format. Wherever possible, context is also separated from content. XML applications represent the progression of the separation of content and form approach. Earlier attempts at separation can be found in HTML documents.

**Separation of Content and Form on the World Wide Web**

When Tim Berners-Lee developed the World Wide Web, he, by necessity, incorporated the principle of the separation of content and form. In those early days, the Internet pipelines were only capable of carrying very small amounts of information, and Web pages cluttered with formatting instructions would not be able to be transferred quickly. So Berners-Lee developed HTML with simple semantics; there were tags for titles, headings, ordered lists, definitions, quotations, citations, paragraphs, addresses, and so on. There was no provision for indents, or fonts, or spacing, or alignment, or columns, or even tabs. Presentational rules embedded in the Web browser determined how a heading would be displayed, how much indent would be applied to lists, what font paragraphs would be displayed in, and what colour text would be used.
During the period known as the "browser wars", Netscape and Microsoft added support in their browsers for new formatting tags, without much consideration to the theoretical underpinning of the Web (and without approval of the World Wide Web Consortium, the custodians of Web standards). Thus, font, colour, and other formatting instructions became mixed up with semantic mark-up, and the benefits of separation were largely lost. It took some years before the World Wide Web Consortium recovered the situation with the introduction of Cascading Style Sheets (CSS). While HTML 1.0 left the formatting entirely up to the browser, CSS allowed the browser to be provided with formatting instructions for an HTML page, separately from the content itself. The content could reside in an HTML file, and the form could reside in a CSS file.

By this time, another limitation of HTML had been uncovered: the semantics were too limited. The solution to this limitation was XML. From a writing perspective, XML reinforced the trend to move to structured authoring approaches.

Structured Authoring

Structured authoring can mean many things, but in the context of this whitepaper, structured authoring means a standardised, methodological approach to content creation incorporating systematic labeling modular, topic-based architecture, constrained writing environments, and the separation of content and form.

The term structured authoring is applied to a wide variety of writing approaches, to the point that the meaning is virtually lost. Some say that most technical writing is "structured authoring" or "structured writing", because the writing process is approached in a methodical, structured way. According to this definition, all documents with some sort of structure must have been the result of a structured approach.

Methodological or scientific approaches to writing technical documents became prominent in the 1960s, with Robert Horn’s structured writing ideas (later to become Information Mapping) and the STOP methodology (developed at Hughes-Fullerton) being two of the intellectual products of that era. The development of SGML (The Standard Generalized Markup Language) almost two decades later enabled structured approaches to be enforced by software tools. The development of XML in the late 1990s transformed the way in which knowledge was stored. XML permitted structured information standards to be created for the storage of knowledge and data for all types of industries. XML allowed standards such as Chemical Mark-up Language, Mathematics Mark-up Language, Channel Definition Format, Scalable Vector Graphics, Open Document Format, and hundreds of others to be created by industry, government and special interest groups.

A modern definition of what we now mean by structured authoring is:

"A standardised methodological approach to the creation of content incorporating information types, systematic use of metadata, XML-based semantic mark-up, modular, topic-based information architecture, a constrained writing environment with software-enforced rules, content reuse, and the separation of content and form."

In the documentation field, new forms of structured writing approaches emerged, enabled by XML and the new culture of the open source movement. Semantic mark-up was the next stepping stone on the path to greater separation of content and form.

Semantic Mark-up

The use of semantic mark-up in DITA, where text elements are marked up based on their meaning, allows the content to be essentially separated from its rendition and display to the reader. For example, a term is marked up as a `<term>` and a citation as a `<cite>`, and no information about how those elements will be displayed is stored in the content. Stylistic (display) rules are applied when the DITA content is transformed into a reading format, such as HTML or ink on paper. In a DITA workflow, documents are created as collections of modular, re-usable topic files, and mechanisms allow not only the format to be separated from the content, but also the context. The same topic may be a section in the context of one publication, but a sub-section in the context of another.

By contrast, the intermingling of content, format and context in a style-based document workflow essentially eliminates the possibility of reuse. Once a paragraph is styled as having a 13 cm left...
Automated processing requires a significant one-off effort to produce the templates that will control the mapping of semantic elements to presentational formatting, but needs very little ongoing effort. A phrase marked up in italic won't render as italic on a reading device that doesn't support italic. But a citation identified as a citation in a DITA topic can be processed to italic by one transformation process, to bold red by a different transformation process, and to synthesized voice by another transformation process.

**Processing**

One of the primary means by which DITA delivers efficiency is through "automated processing", or "transformation".

"Automated processing" means that the publishing process (transforming semantically marked-up source content into a reading format) is wholly automated. Any valid DITA document should be able to be processed in exactly the same way. Automated processing requires a significant one-off effort to produce the templates that will control the mapping of semantic elements to presentational formatting, but needs very little ongoing effort.

In other words, publishing templates are created to suit the formatting and layout requirement standards for a company. Once those templates are complete, they are used, without further intervention, for all that company's publishing needs, forever. These templates are based on the company's House Style guidelines. They lock in the presentational style rules.

If each manual produced by a company needs a different presentational style, then automated processing becomes "semi-automatic". It means that templates have to be developed for each manual. Instead of being a one-off effort, the significant cost of creating templates becomes a recurring cost.

There are consequential benefits of automated processing. Usability is improved by automated processing, because consistency is a key component of usability. Users find it more difficult to work with a manual if it looks different to other manuals. Consistency also leads to authority; users subconsciously associate "inconsistent presentation" with "inaccurate information".

If a document is layout-intensive, it is less able to be processed automatically. For example, if a diagram in the left column needs to be lined up with particular paragraphs in the right column, the mark-up cannot be done in isolation from the publishing process. Another way of putting it is that the separation of content and form becomes difficult, and that separation of content and form is what allows:

- reduced cost through automated processing
- reduced translation cost
- reduced cost through content reuse
- value-adding through multi-channel publishing (producing output in many formats)

To explain the impact on content reuse and single-sourcing, we can use the example of the diagram in the left column aligning with text in the right column. DITA does permit elements to be marked up with attributes that allow them to be treated specially during processing. We can use these output attributes to define our column breaks, but this means we end up with paragraphs that can only be practically used in the context of a two-column layout with the diagram to the left. This means we cannot reuse some of those paragraphs in different places in the manual (or in a different but similar manual), because the content relies on the same diagram being positioned to the left of the paragraph. Further, we cannot generate the same content in HTML format for publishing on the Web, because Web browsers “re-flow” the text to suit the user’s window size, and HTML doesn’t support flow through columns. In other words, the extra effort expended to make the layout work results in the content being less useful!

In practice, it is very difficult to completely separate content and form. Consider a table, for example. A table is a formatting construct on the one hand (a method of dividing some types of content into columns), and a semantic structure for storing reference information (as in database tables). It is hard to create a table without concern for how it might be presented. Likewise, an image is both content and form together. The image file contains colour, but also contains data.

**Distinction between Format and Style, and Data and Metadata**

In DITA, format has a different meaning to style. Likewise, the meanings of data and metadata are very different. These distinctions in meaning are important to understanding the broader concept of the separation of content and form.
The word "style", as in "Style Guide", is problematic, because it has a number of subtly different meanings in this context. Style could mean aesthetic presentational style, and it could mean writing or wording style. A style could be clean and crisp (aesthetically) while being ponderous and wordy (stylistically).

To distinguish between the two "styles", format (or presentational style) should be used when referring to aesthetic style, or the look and feel of the deliverable document. The term writing style should be used when referring to the authorial style. In the broader concept of the separation of content and form, writing style belongs to content, while format belongs to form.

It is also important to understand the distinction between data and metadata. Data is analogous to content, while metadata refers to information about the content.

For example, the data of a topic is the information that the reader reads about the subject matter of the topic. The metadata is the supporting information about the topic that the reader doesn’t normally read or see, such as the creation date, the author, the semantics of the textual components, and the copyright ownership. Metadata is critical to separation of content and form, as it stores information that is important for document processing.

Challenges for Technical Communicators

The philosophical differences between style-based authoring and semantic authoring present the greatest challenges for technical communicators.

In style-based authoring, the structure of a document is defined by the styles applied to its components. For example, a second-level heading might be defined in a document by the application of a Heading 2 style. Likewise, the presentational style of the deliverable document is defined by styles embedded within the authored document. For example, text to be indented by 2 cm might be defined in the paragraph by the application of an Indent 2 style.

The separation of content and form in DITA sees the upper-level structure being defined outside the content (in the ditamap), and the presentational style being applied in a publishing process entirely separate from the authoring process. The presentational form for a document can be unknown to the DITA author. The same topic can have different appearances and different structures when the same source content is used to produce different deliverable documents.

For example, a DITA topic in one deliverable document may have a Heading 2 style applied during processing, but the same topic in a different deliverable document may have Heading 4 applied. The publishing rules determine the mapping of DITA semantic elements to output presentational form.

Techniques to Learn

Authors moving from style-based authoring to structured authoring will need to build chunking, labeling and linking skills, and to embrace the technique of separation of content and form.

DITA authoring in particular requires you to have skills that you may have used in style-based authoring, but which you will certainly need to use differently. Some skills may be entirely new to you. These skills are:

- **Chunking**
  Chunking refers to the way in which you break down information into smaller pieces. The term is particularly (but not exclusively) used to describe the way in which information is broadly categorised into information types, or topic types.

- **Labelling (or metadata creation)**
  Labels and catalogue information are part of a topic’s or collection’s metadata. Metadata allows content to be filtered, sorted, processed, and otherwise manipulated. Choosing accurate labels will result in more flexible documents.

- **Linking**
  Linking can be viewed as a technique for defining relationships between topics.

- **Separating content from form**
  In the writing phase of structured authoring, there is no place for form (format, style and presentation). Form is not the author’s job. Identifying the text’s semantics is.
Separation, not Removal

Many objections to the separation ideal rightly point out that format is a vital component of communication. However, "separation" is not the same as "removal". Formatting of the content is indeed a very critical part of effective documentation. Separation may actually improve the quality of formatting, because it makes it easier for an expert in form, such as a graphic designer, to work alongside an expert in writing.

It is true, however, that content and form can seldom be cleanly divided. As Karen McGrane pointed out in the article "WYSIWTF on A List Apart"7:

"Arguing for separation of content from presentation implies a neat division between the two. The reality, of course, is that content and form, structure and style, can never be fully separated."

Conclusion

Changing focus from form to content may sound easy, but it is turns out to be quite a difficult transition... writers are so accustomed to working with mixed content and form. The benefits in separation are many. By automating the formatting process, technical writers can spend more time on the words and phrasing, rather than on the fonts, alignment and numbering! This leads to better writing quality, and more consistent presentation.

The advent of a myriad of new reading devices, such as head-up displays in spectacles and pico-projectors, is making separation of content and form an indispensable tool in single-sourced, modular, cost-effective documentation.

Author Biography

Dr. Tony Self is the co-founder of HyperWrite, an Australian hypertext solution company, founded in 1993. HyperWrite provides consultancy and training services focused on Help systems, technical writing technologies, and online documentation strategy. Tony has worked as a technical communicator for over 30 years, with the last 20 of those years specifically in the areas of online Help systems, computer-based training, and electronic documents. The majority of his work currently involves providing training and consultancy in structured authoring, Help systems, DITA, and technology strategy. Tony completed his PhD in semantic mark-up languages in 2011, and his book “The DITA Style Guide” was published in the same year. He also holds a Graduate Certificate in Teaching and Learning and a Graduate Diploma in Technical Communication, and is a Fellow of the Institute of Scientific and Technical Communicators (UK). He is a member of the OASIS DITA Technical Committee (and chair of the DITA Help Subcommittee), and is an adjunct teaching fellow at Swinburne University. Tony won the ISTC Horace Hockley award in 2011.