34. Structure

Carlos Evia
Virginia Tech

The implementation of structural moves in ***public*** and professional discourse has been practiced and studied in disciplines related to ***rhetoric*** for centuries. In traditional rhetorical treatises, for example, persuasive speeches were not presented as amorphous sequences of words. Instead, they were assembled according to specific structures that could include sections announcing, explaining, outlining, supporting, and summarizing the parts of a speech (Cicero, 2014). Each component of those structures served a purpose that eventually enabled the production of specific modes of discourse beyond persuasive oratory. This process of identifying, documenting, and implementing common structures to preserve order and rules in discourse established a longstanding tradition of applied rhetoric in writing studies. Technical and professional communication continues this tradition, as some of its scholars acknowledge that "many discourse conventions are, in fact, formalizations of rhetorical moves" (Flower, 1989, p. 34).

For technical communicators, the formalization of rhetorical moves into common structures enabled the production of ***genre***-based documents. Instead of a disjointed collection of paragraphs, a written ***proposal*** can have structural components that *propose* something to a specific audience, a report can have structural sections that *report* on a situation for interested readers, and a set of instructions can *instruct* users on how to accomplish a series of tasks. The production of technical communication genres with established conventions and expected components established the field's importance in the computing industry, as corporate and academic authors published guidelines for structuring technical ***documentation*** and manuals for software and hardware in the 1970s and 1980s (Cohen & Cunningham, 1984; Price, 1984; Rigo, 1976).

Applied at a presentational level, structures in technical communication can create content templates, which have been described as "a kind of wizard for content development" (Kissane, 2009). Content templates can establish that, for example, every section in a quick start guide for a new computer *should* have a title, a paragraph, and a numbered list. Content templates can be implemented as formatting structures in most desktop publishing software applications or, for online publication, with presentational tags from Hypertext Markup Language (HTML), which is foundational to most web-aimed ***content management*** solutions.

For technical communicators, the main benefit of using templates to structure content is the availability of pre-determined styles. Writers "don't spend time

DOI: https://doi.org/10.37514/TPC-B.2023.1923.2.34

figuring out how to create particular formatting—they apply ***styles*** to add formatting" (Pringle & O'Keefe, 2009, p. 41). For consumers of technical content created with a template, the main benefits are defined structural patterns that keep content consistent and make it easier to skim or browse.

At a semantic level, structure supports the practice known as structured authoring, which is "a publishing workflow that lets you define and enforce consistent organization of ***information*** in documents, whether printed or online" (O'Keefe & Pringle, 2017, p. 2). Beyond what a template can establish, structured authoring dictates that content *must* adhere to a specific structure. Structured content "clearly indicates not only the parts of the discourse (the titles, sections, lists, tables, and phrases that represent organization) but also the semantic intent of those containers" (Day, 2016, p. 51). Therefore, if a template allows formatting of a quick start guide, structured authoring can specify that the title is a section heading, the paragraph is an introduction, and the numbered list is a series of steps (see Figure 34.1).

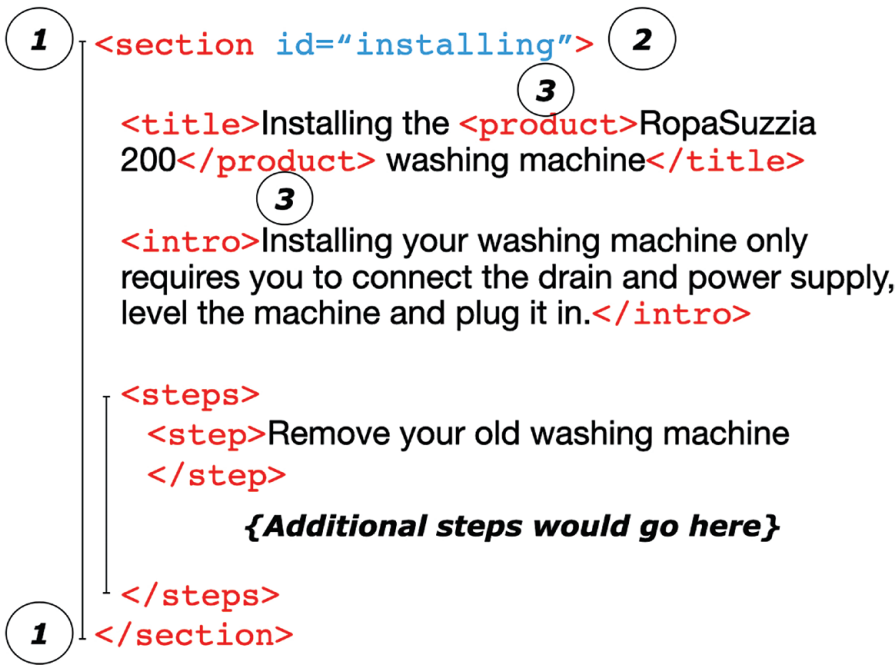

Figure 34.1. Structured section from a fictional quick start guide for a washing machine. No. 1 shows markup tags (using Extensible Markup Language) that describe the code snippet as a section in what could be a larger guide. The opening bracket for the section opens at the beginning of the snippet and closes at the end. No. 2 shows an attribute that gives a unique identifier to the section. The section element contains block sub-elements for title, introduction, and steps. No. 3 shows an inline sub-element referencing the product name.

For the past two decades, structured authoring in technical communication has frequently involved implementations of the Extensible Markup Language (XML). Particularly, the Darwin Information Typing Architecture (DITA) has become one of the main XML grammars used for technical communication purposes. DITA started as "a technical documentation authoring and publishing architecture that is based on principles of modular reuse and extensibility" (Priestley et al., 2001, p. 352) at IBM. Since 2004, DITA has been an open standard maintained by the nonprofit Organization for the Advancement of Structured Information Standards (OASIS).

The modular structure of DITA is based on a generic topic type that can describe almost any content. In a DITA authoring environment, writers create "technical content by assembling topic-oriented information types or blocks of information that serve particular functions in a document" (Swarts, 2010, p. 133). Over the years, the DITA standard has specialized the generic topic into information types that "represent the vast majority of content produced to support users of technical information" (Hackos, 2011, p. 7). These topic types for structuring technical content include concept, task, reference, glossary, and troubleshooting. The DITA standard also includes topic types designed for structuring learning and training projects: learning plan, learning overview, learning content, learning summary, and learning assessment.

Besides the preestablished topic types for technical content and learning and training projects, DITA topics can be customized (in a process known as specialization) to create information types unique to any domain. This exercise in markup flexibility is a direct application of both the *extensible* part of XML and the *Darwin* element in DITA: XML elements can be *extended*, and DITA information types can *evolve* to structure diverse content needs. For example, a DITA specialization for music composition could have a topic type for song with predetermined elements for intro, chorus, and bridge.

For technical authors, potential benefits of structured authoring in a workflow using DITA (or a similar standard) include streamlining the content creation process, increasing the quality of content by standardizing it, and allowing authors to leverage content in many different ways, which include reusing it, publishing it in different formats, and translating it (Samuels, 2014). From a business perspective, DITA can lead to promoting the reuse of information quickly and easily across multiple deliverables, which leads to reducing the cost of maintaining, updating, and localizing information (Hackos, 2011).

The reuse capabilities of structured content are the strongest selling points of a standard like DITA. Kristen Eberlein (2016), chair of the DITA Technical Committee with OASIS, defines reuse as the "practice of using content components in multiple information products" (p. 54). She adds that in many technical communication workflows, "efficient content reuse does not involve copy-and-pasting; instead it uses transclusion, whereby content is authored in one location and used by reference in other locations" (Eberlein, 2016, p. 55)

Content structured in DITA or DITA-like methodologies also opens the possibility of single sourcing, which can be defined as the practice of "creating content once, planning for its reuse in multiple places, contexts, and output channels" (White, 2016 p. 56). The tags of an XML-based grammar like DITA also can make content behave like data; as a result, structured content is computable and allows machine processing (Day, 2016 p. 50). Structured content can include metadata, which is defined as "'data about data,' which means data that isn't the primary purpose of the content object, but serves some secondary purpose" (Barker, 2016, p. 92). With an appropriate combination of structure and metadata, for example, the same task on how to configure a new computer can include introductory steps for inexperienced users and advanced steps for expert users (e.g., <*step audience="introductory"*> *Turn on the computer*</*step*> and <*step audience="advanced"*> *Replace the motherboard*</*step*> can be included on the same task). Publishing instructions would then include filters and routines to produce deliverables aimed at either inexperienced or advanced users that occlude (but do not delete from the structured source) content that would be irrelevant for the intended audience group.

Despite its actual and potential benefits for content creators and their business supervisors, the implementation and enforcement of structure in technical communication authoring workflows is not without its challenges. A major challenge to widespread adoption is the separation of content and presentation required by workflows based on DITA or a similar standard. This separation "can create philosophical and cognitive dissonance for technical communicators trained to think of information as content that is inherently linked to presentation" (Clark, 2007, p. 36). According to some, writers separating content from presentation "will have no control over the context in which their information appears or the uses to which it may be put" (Gu & Pullman, 2009, p. 6). Adopting templates in desktop publishing applications could, therefore, be an effective introduction to structure for novice technical communication practitioners, and it might be enough for situations in which content reuse and single sourcing are not required. If the reuse needs of a project change or evolve, commercial and open-source tools can relatively easily convert template-based documents to structured content using DITA or a similar standard.

Another challenge is the perceived loss of creativity for authors using a structured content type as opposed to a writing environment without restrictions, or "the perception that XML forces writers into creating cookie-cutter topics rather than useful technical information" (O'Keefe, 2010, p. 37). Taken to its most dangerous extreme, the implementation of structure in technical communication could lead to the standardization of cultural products that Theodor Adorno (1991) presaged. However, taken to its most beneficial extreme, structured content workflows could produce information schemas like those proposed by J.C.R. Licklider (1965) for cataloging cultural artifacts, which revolutionized the ways in which ***technology*** helps librarians gather, index, organize, store, and distribute print and digital content. Some scholars tackle this challenge as an opportunity

to acknowledge that "while the technology can hamper some elements of creativity, it can also open up new possibilities for rhetorical expression, for writing content that can be assembled into new meaningful forms" (Swarts, 2020, p. 171).

The evolution of structure in technical communication is leading to the development of more flexible methodologies and standards (e.g., Markdown, JSON, and proprietary solutions for separating content from presentation). Although they do not provide all the capabilities of XML, they can replicate most of the transclusion and single sourcing features of DITA (Evia et al., 2018). Evolutionary trends also include the practice of content-as-a-service (CaaS), which "focuses on managing structured content into feeds that other applications and properties can consume" ([A], 2017). In a CaaS-based workflow, structured content does not necessarily inherit formatting and processing rules from the same organization where it is developed, but it is available for use in different contexts and environments via online information requests.

As an explicit change of tone in speech or a new section in a piece of user documentation, or behind the scenes as a command for a computer request sending content to a voice application, structure is essential to technical communication. Audiences and authors will continue evolving, and their use of technology will no doubt become more sophisticated and complex over time. Regardless of medium and technology, a well-structured document will always be a more effective piece of communication than a disorganized blob of words.

References

[A]. (2017, March 2). *CaaS: What is content-as-a-service?* https://simplea.com/Articles/what-is-content-as-a-service

Adorno, T. W. (1991). *The culture industry: Selected essays on mass culture.* Routledge.

Barker, D. (2016). *Web content management: Systems, features, and best practices*. O'Reilly.

Cicero, M. T. (2014). *De inventione; De optimo genere oratorum; Topica*. Harvard University Press.

Clark, D. (2007). Content management and the separation of presentation and content. *Technical Communication Quarterly*, *17*(1), 35-60. https://doi.org/10.1080/10572250701588624

Cohen, G., & Cunningham, D. H. (1984). *Creating technical manuals: A step-by-step approach to writing user-friendly instructions*. McGraw-Hill.

Day, D. (2016). Structured content. In R. Gallon (Ed.), *The language of technical communication* (pp. 50-51). XML Press.

Eberlein, K. J. (2016). Content reuse. In R. Gallon (Ed.), *The language of technical communication* (pp. 54-55). Laguna Hills, CA: XML Press.

Evia, C., Eberlein, K., & Houser, A. (2018). *Lightweight DITA: An introduction. Version 1.0*. OASIS.

Flower, L. (1989). Rhetorical problem solving: Cognition and professional writing. In M. Kogen (Ed.), *Writing in the business professions* (pp. 3-36). National Council of Teachers of English.

Gu, B., & Pullman, G. (2009). Introduction: Mapping out the key parameters of content management. In G. Pullman & B. Gu (Eds.), *Content management: Bridging the gap between theory and practice* (pp. 1-12). Baywood.

Hackos, J. T. (2011). *Introduction to DITA: A user guide to the Darwin Information Typing Architecture including DITA 1.2* (2nd ed.). Comtech Services, Inc.

Kissane, E. (2009, July 7). Content templates to the rescue. *A List Apart*. https://alistapart.com/article/content-templates-to-the-rescue/

Licklider, J.C.R. (1965). *Libraries of the future*. MIT Press.

O'Keefe, S. (2010). XML: The death of creativity in technical writing? *Intercom*, *57*(2), 36-37.

O'Keefe, S., & Pringle, A. (2017). *Structured authoring and XML*. Scriptorium.

Price, J. (1984). *How to write a computer manual: A handbook of software documentation*. Benjamin Cummings.

Priestley, M., Hargis, G., & Carpenter, S. (2001). DITA: An XML-based technical documentation authoring and publishing architecture. *Technical Communication*, *48*(3), 352-367.

Pringle, A., & O'Keefe, S. (2009). *Technical writing 101: A real-world guide to planning and writing technical content*. Scriptorium Press.

Rigo, J. (1976). User manual outline. *Asterisk*, *2*(8), 7-10.

Samuels, J. (2014, February 3). What is DITA? *TechWhirl*. http://techwhirl.com/what-is-dita/

Swarts, J. (2010). Recycled writing: Assembling actor networks from reusable content. *Journal of Business and Technical Communication*, *24*(2), 127-163. https://doi.org/10.1177/1050651909353307

Swarts, J. (2020). Writing about structure in DITA. In T. Bridgeford (Ed.), *Teaching content management in technical and professional communication* (pp. 155-173). Routledge. https://doi.org/10.4324/9780429059612-9

White, L. W. (2016). Single sourcing. In R. Gallon (Ed.), *The language of technical communication* (pp. 54-55). XML Press.