What is structured content?

Stanley Doherty, Ph.D.

Guidepost

Purpose: This resource introduces structured content and the various ways that you can work with it.

Learning objectives:

- Differentiate content from formatting.
- Understand unstructured content.
- Understand informal structured content.
- Understand formal structured content.
- Review additional learning resources.

Content, context, and formatting

When we write a document for a particular class or project, we just want to capture the content. We format (style) it as we go – headings, lists, and fonts. The formatting supports the delivery of that content in one context.

This HTML markup …

```
<p>The French Revolution espoused <b>Liberté</b>, <b>Egalité</b>, and
<b>Fraternité</b>.</p>
```

… gets formatted into . . .

The French Revolution espoused Liberté, Egalité, and Fraternité.

There is a 1:1 relationship between the content markup (`<b>Liberté</b>`) and its output formatting (Liberté).

What happens if we want to use the same content in multiple contexts, each with a different presentation format or behavior? Consider more descriptive markup.

The markup …

```
<p>The French Revolution espoused <gloss>Liberté</gloss>,
<gloss>Egalité</gloss>, and <gloss>Fraternité</gloss>.</p>
```

… can still produce . . .

The French Revolution espoused Liberté, Egalité, and Fraternité.

But the more descriptive element `<gloss>` can also be transformed for multiple outputs (contexts) – pop-up glossary links, index entries, or input for AI prompts.

When we decouple content from its immediate context and formatting, we can transform it and reuse it in many ways. It becomes "intelligent."

Unstructured content

The vast majority of all content is unstructured:

- Research papers
- Business reports
- Social media posts

In these documents, the sequence of elements such as headings, lists, and paragraphs flows from the meaning that you want to convey – not from some separate template. Each unstructured content document is unique, a special snowflake.

If you do not need consistency or flexibility in reusing content in multiple contexts, unstructured content will do the job.

Sample markup

The following HTML file is not derived from a template. It is freeform.

```
<!DOCTYPE html>
<html>
  <body>
```

```
    <h3>Requirements</h3>
      <p>There are many requirements to be an astronaut.</p>

    <h2>Astronaut training</h2>
      <p>There are many requirements to be an astronaut.</p>

  </body>
</html>
```

HTML web browsers reading this file are permissive and would not care that an <h3> precedes an <h2> heading.

Structured content - in general

Organizations turn to structured content when they need to produce many thousands or millions of documents that are highly consistent or valid.

For example:

- Technology companies want to produce documents with consistent formatting and navigation to support a positive customer experience.

- Regulated medical companies need valid data to support analysis.

To produce structured content, you need to have a clear picture of the building blocks of content that you want *and* how you want to organize them. It is best to figure this out *before* you create a thousand instances. It all starts with a clear content model.

Templates, schema, and document type definitions are your new friends.

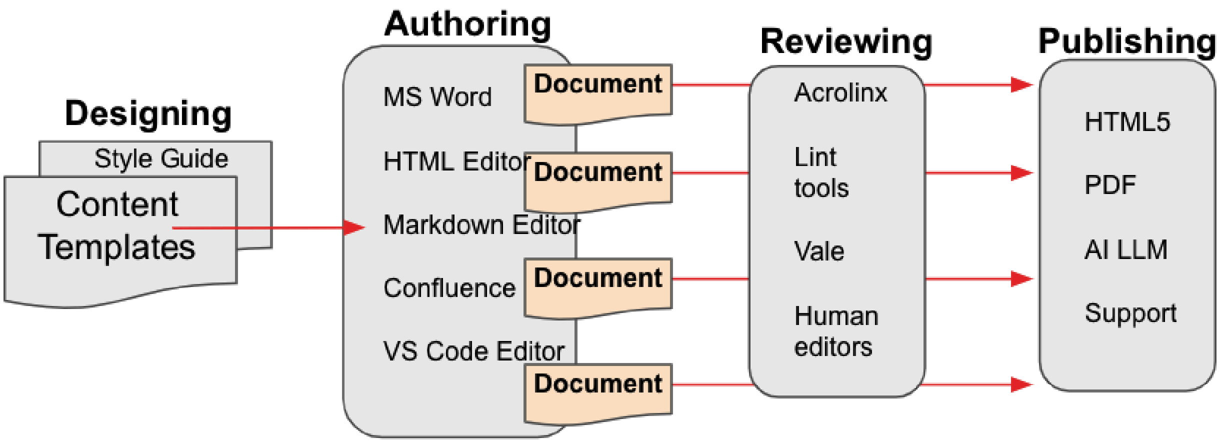
Informal structured content

Organizations can achieve serious consistency in their topics/pages if they deploy:

- Content templates for each author - specifying desired content building blocks, usage, and assembly best practices.

- Automated checkers - content profiling tools that have rules for evaluating the compliance of any one document to content templates and standards.

- Human editors - spot checking markup and usage issues.



Sample markup

The following Markdown file was created from a content template.

```
<!-- Created from our-concept.md v2.2 template. –>
# Astronaut training
To become an astronaut, you'll need a thorough background in engineering and
years of flight experience.

## Astronaut requirements
* Masters in Engineering
* 700-800 hours of certified flight time

See [NASA application](http://nasa-forms.html) for information.
```

If we ask an automated checker application to review these files, it will read the declaration on line-1 about the `our-concept.md` template and begin checking this file for its compliance with that Markdown template and any other usage rules that we require.

Formal structured content

When organizations need to have every document comply with a content model, they rely on strict document definitions and validating parsers.

- **Document definitions**: Specify the order, containment, and values required for content elements in an instance of an XML DITA document type. In a DITA task topic, for example, you do not see separate <section> elements as you might in a concept or reference topic. These document definitions come in forms: XML (DTDs), databases (database table definitions), and JSON (schema files). When we create a piece of formal structured content, we first choose its document definition.

- **Validating parsers**: Evaluate whether a particular instance of a document or database table conforms exactly to the document definition that you select when you create it. If I use the schema.org document definition for a JSON book description, the validating parser compares the JSON document on my desktop to the document definition stored on schema.org. If my document and the document definition match 100%, the document is valid. Otherwise it is invalid.

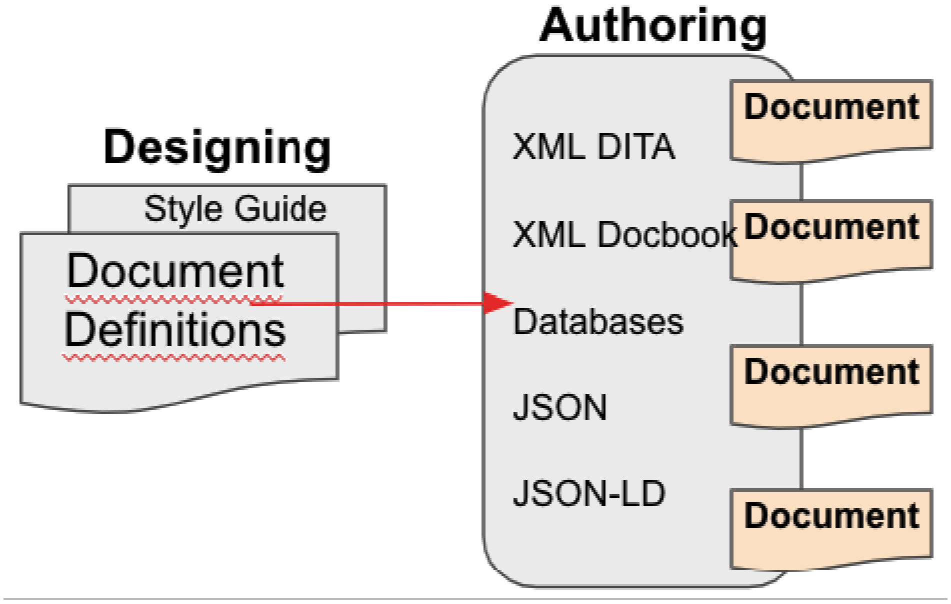
Scenario

Let's walk through a typical workflow for a software documentation project using formal structured authoring.

1. Information architects in a company create a content model and a collection of supporting document definitions for the content that needs to be delivered.

2. Authors download these document definitions into their editor/author software application.
   - `our-task.dtd`, `our-map.dtd`, `our-concept.dtd`, and `our-faq.dtd`

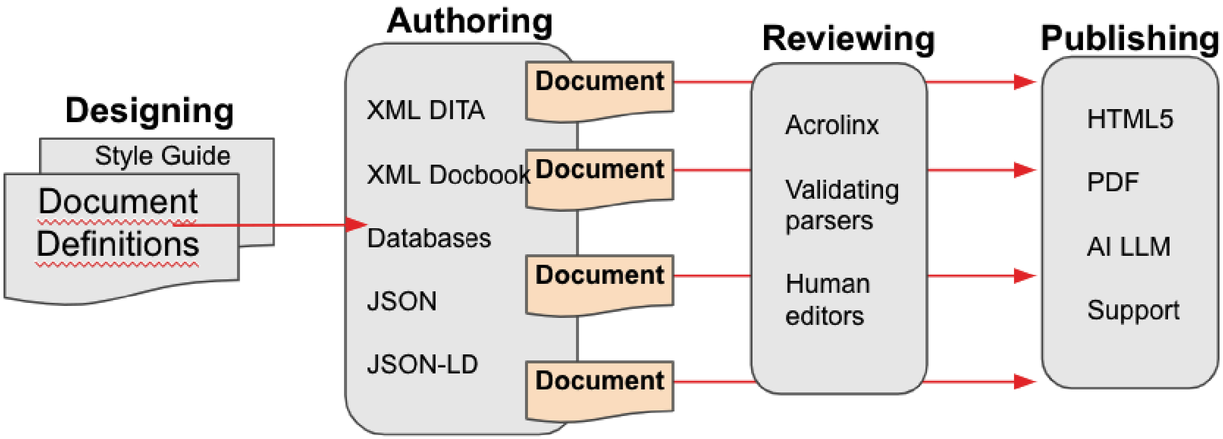
3. As they develop new topics, authors choose **File - New** and specify a particular DTD to use.



4. As authors add elements and text to a new topic, the validating parser built into an XML or JSON editor compares what is in the new topic to its corresponding DTD. If an author creates an instance of `our-faq.dtd` and attempts to insert an element that is not supported by `our-faq.dtd`, the parser generates a validation error.

5. Authors then check their valid topics into GitHub or into a content management system.

6. Automated build systems collect all the files, re-validate them, and publish them.



Sample markup

This XML DITA 1.3 topic was created from the concept.dtd document definition.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept>
    <title>Astronaut training</title>
    <conbody>
        <p>To become an astronaut, you'll need a thorough background
           in engineering and years of flight experience. </p>
        <p>See <xref href="http://nasa-forms.html" scope="external"
              format="html">NASA application</xref> for more information. </p>
    </conbody>
</concept>
```

When a validating parser reads line-2, it compares everything in this topic to the rules in the `concept.dtd` document definition. Any deviations in this document from `concept.dtd` cause the parser to report this topic as invalid. It will not build.

ACM Learning Resources

- Oxygen Editor Quickstart - Creating, Assembling, and Building a DITA Publication
- Oxygen Editor Quickstart - Configuring Layout and Preferences
- Heretto CCMS Quickstart - Creating, Assembling, and Building a DITA Publication
- DITA Quickstart (MacOS) - Creating, Assembling, and Building a DITA Publication with a Text Editor and the DITA Open Toolkit
- DITA Quickstart (Windows) - Creating, Assembling, and Building a DITA Publication with a Text Editor and the DITA Open Toolkit

Conclusion

Gaining familiarity with multiple flavors of formal and informal structured content has many benefits:

- It allows you to develop a more comprehensive portfolio of writing samples.
- It reduces the amount of training that you would need in joining a team working in structured content. This matters to hiring managers.
- It provides you with some insight into converting content between tools and architectures.
- It provides a sophisticated vocabulary of concepts to discuss in interviews.

Thank You!

See https://acm-sigdoc-structured.org to learn more about committee activities, available resources, and volunteer opportunities.

Created by members of the ACM SIGDOC Committee on Structured Authoring and Content Management