Like Beads On a String: Structured, Topic-Based Authoring in Flare
Who Am I?

  - Internationally recognized content creation and delivery consultant.
  - Help clients create effective, efficient, flexible content in anything from print to mobile.
  - Certified – Flare, other tools.
  - Creating structured content in topic form since 1986, with Flare since 2006.
  - Worked with DITA since ‘02, former member of OASIS DITA sub-group.
What Is Topic-Based Authoring?

- Authoring content in chunks – “topics” – rather than documents or books.
- Not new.
  - Dates back to ‘91 with first HATs.
  - Actually dates back to ‘65 and InfoMapping.
What’s a Topic?

- As **focused** and **self-contained** *as possible* a discussion about one subject.
  - **Focused** – Answers one question – “How do I...?”, “What is...?”, etc.
    » Not DITA, though conceptually similar.
  - **Self-contained** – Contains all info needed to answer the question.
    » Related info is in separate topics.
    » Topics link to separate but related topics.
Documents vs. Topics

- A document is one big chunk of content.
  - Difficult or impossible to subdivide and reassemble in different forms.

- A topic is one little chunk of content.
  - Many topics can be threaded together to create the document and other outputs.
  - Remember that necklace you made for mom in summer camp...
What Is Structured Authoring?

◆ Authoring with structure.
  – Just means content has structure – no preconceptions about what it is or how it’s applied.

◆ So what is “structure”?
  – Standard and consistent sectional, syntactical, and stylistic rules.
  – What technical communicators have always created, albeit usually manually…

◆ Two types.
1 – Visually Structured

- Text in Normal style, formatted using the formatting toolbar.
  - A silly example, but often seen as “structured.”
  - A hurdle on the way to structured authoring.
  - Is this structured?
Using styles, perhaps a CSS.

- Head *styles* add *programmatic* structure.
- But there are two problems:
  - No defined structural sequence of the subsections.
  - No enforcement of any structure.
- Is this structured?
An Improvement On #2

- Using a topic template as well as a CSS.
  - The template defines the structure/sequence of the topic’s content.
  - The CSS automatically formats entries in the template.
  - Still no enforcement of the structure though.
    » We’ll get to that at the end.
What Do I Get Out of This?

- Flexibility – Granular content easier to target.
- Multi-author capability – Multiple authors can work on portions of the same project.
- Re-use – Apply chunks to multiple projects.
- Consistency/accuracy – Writing a chunk once = consistency, fewer errors.
- Ownership – One owner per chunk.
- CMS integratibility – Self-explanatory.
- Participation in company’s strategic definition?
But…

◆ It’s a new way of authoring.
  – No continuity between topics makes it hard for authors to stay focused.
  – No real-world analog makes it hard for authors.
  – Hard to re-use content for stylistically different outputs, like tech support and marketing.
  – Editing many topic variations = MEGO.
But...

- And can be hard to manage.
  - A topic should be the smallest unit of content that answers a question.
  - If you create very granular topics, like making each field description a topic, the number of topics can become unmanageable.
  - May be hard to communicate project detail to the next generation of developers.

- KISS!
Structured Topic-Based Authoring Features in Flare
Flare is innately topic-based.

Button

This feature, on the Design screen, creates a button with a text label, such as Save Entry.

Here's the button properties dialog box.

Internal Button Name is the name by which ViziApps refers to the field programmatically. This entry is required.
Topic-Based Authoring Features

- But without structure, those topics are just inconsistent piles of text.
- How do we add “structure” thru Flare?
Topic Templates

- Flare ships with a set of predefined topic templates.
- But you can add your own custom templates as well.
Topic Templates

- Templates let you define the structure, common content of specific topic types – concept, task, reference, troubleshooting, etc.
- To create a topic, you now categorize it, select the appropriate template, and then start writing.
  - Topics based on a template automatically use its structure, common content.
  - You can focus on topic-specific material.
Topic Templates

◆ You can define one template for each topic type and send them to all authors for consistency and to do “structured” authoring without DITA.
Why Templates Matter

◆ For authors:
  – Help create consistent, predictably structured, topic-based material.
  – Eliminate time wasted trying to decide how to structure a document.
  – Support a possible future shift to DITA.

◆ For users:
  – Consistently, predictably structured material is easier to understand and use.
Additional Template Benefits

- You can assign styles to elements of a template.
- When you pull the template into a project with a master CSS, topics automatically get structured and formatted.
- This is structured authoring, albeit with human enforcement.
Sample “Task” Template

[delete and type the title]

[delete and type the intro description]

Date of Applicability

[delete and type the date]

Required Materials

[delete and type the tools and materials list]

and so on…
Adding Templates To Flare

- Add your templates to Flare’s interface using the Template Manager – Tools > Manage Templates.
Defining Information Types

- Defining info types is step 1 toward templates.  
  - Means identifying types of content you create and putting them into *a few* standard categories:
    » Concept, task, reference, troubleshooting, show-me, process, procedure, others.

- If some material doesn’t fit into the types:
  - Could it fit with some modification?
  - Does it call for a new type?
  - Is it just weird?
Mining Templates for CSS Elements

- Elements in your templates – H1, H2, bulleted lists, etc. – are also the elements you must define in the CSS.
- Keep a list of those elements as you create your templates.
- Do you need custom elements – “Note” or “Tip” – or will standard CSS elements suffice?
CSS

- A file containing the format settings for your topics.
- Once you apply styles from a CSS to text, you can change all instances of a style by changing its properties once in the CSS.
- Moving all formatting information out of topics and into a CSS follows W3C best practices and helps future-proof your content.
Why Use CSS?

- Efficiency – Define all attributes of each style once in the CSS and link each HTM file and topic template to that CSS.
- Consistency – Having other authors use the same CSS standardizes the formatting.
  - If all authors use the tags consistently, which suggests creating and using topic templates.
Table Styles

- A CSS specifically for table formatting.
- Provide efficiency, consistency like a regular CSS.
- Create using the Tablestyle Editor.
Other “Structure” Features

- Project Import – Lets you put standard files in a master project and use those files in child projects live linked to the master project.
  - To use the same templates or CSS in multiple projects – cross-project structural control.
  - Provides “hidden” control – both powerful and simple but must be documented.
Writing, Writing Support Features

- **Topic-based writing in one bullet** – There is no more continuity; what used to be a section may now be a topic so write self-containedly, replace “as described above/below” with links.

- **Variables and snippets** – Use for repeated bits of content but also to ensure consistent phrasing of content – e.g. Digital Equipment Corp. vs DEC.

- **Analyzer** – Can search projects for content that could be turned into variables or snippets.
Where’s DITA In All This?

- Don’t we need DITA to do structured authoring?
  - No. Flare’s native features let you create topics with structure without using DITA.
  - What Flare does not do is programmatically enforce that structure.
  - If you need that, then you need DITA.
  - What are the trade-offs?
Flare/DITA Trade-offs

◆ Flare provides:
  – Conceptual and technical simplicity.
  – Authorial flexibility – you make the rules (but have to enforce them).
  – A very short startup analysis and design period.

◆ DITA provides:
  – Rigid open-source rules and extensibility under those rules.
Some Final Thoughts

◆ Look beyond the mechanics of structured, topic-based authoring to understand how it fits into the larger business and strategic direction.

◆ Develop in-house expertise.
  – Budget for training and conferences.

◆ “Stay between the lines.”
  – Use tools correctly and avoid “hacks”.
Hyper/Word Services Offers...

Training • Consulting • Development

➢ Flare • Advanced Flare (thru MadCap)
➢ ViziApps
➢ Single sourcing • Structured authoring
Thanks for attending today’s webinar!

As a webinar attendee, receive $100 off any MadCap Training Course:

Discount Code JAN100*

For available training courses and to receive your discount, contact:

sales@MadCapSoftware.com
+1 858.320.0387 opt. 1

*Valid for any current or future training course reserved by February 14, 2014. Certain restrictions apply.
Thank you... Questions?

978-657-5464
nperlin@nperlin.cnc.net
www.hyperword.com
Twitter: NeilEric