To: Chris Lindgren

From: Jacqueline LaLiberte

Subject: DITA topic model design rationale

Date: 12/23/2015

[DITA-project repository](https://github.com/lalib035/DITA-project).

This topic model discusses tasks, concepts, and reference material related to generic containers in Hypertext Markup Language (HTML). For users without a firm grasp of the relationship between Cascading Style Sheets (CSS) and HTML or the importance of semantics, it can be difficult to use HTML’s generic containers, <span> and <div>, properly. The primary goal of this topic model is to assist users in understanding and using <span> and <div> in their digital writing process. This two-pronged goal focuses on the immediate purpose of showing users how to properly use <span> and <div>, and the overarching purpose of bettering users’ general understanding of generic containers, HTML, and CSS.

Three task topics direct users toward this goal. The first task topic, “Choosing a Generic Container,” is a supertask topic. This topic emphasizes the differences between <span> and <div> for users who may be unaware that there is a difference. It also provides a scannable launch point for users who already understand the basic background of CSS, HTML, and web semantics. As a supertask, “Choosing a Generic Container” contains two child task topics: “Using the Generic Container <span>” and “Using the Generic Container <div>.” The short descriptions for these child tasks are displayed in the supertask output and act as user guideposts for determining what task to tackle next.

The task topics on using <div> and <span> share reusable content. This architecture stems from the choice collection type used in the supertask—“choice” implies users will read either the <span> or the <div> task, not both. Because users will not necessarily see both topic tasks, the prereq information should be reused on each task page. The prereq information contained in the reused content is not appropriate for the supertask; however, it is information necessary to complete the child tasks.

While writing the task topics, I considered using a single <result> element at the end of each topic instead of <stepxmp> elements for every step. However, I decided that a single <result> element would be less helpful to users who may not understand what happens in each step. By deconstructing each addition using the <stepxmp> element, I hope that potentially confused readers can better identify the changes made during each step of the process. If this manual came with a help-line phone number or email, this would also allow help associates to identify where things went wrong more quickly.

I also wrote two concept topics to support the primary user goal. The first concept topic, “HTML and CSS,” gives a brief overview of HTML, CSS, and their relationship. While I was writing the HTML/CSS overview, I realized that I was writing content that could be used in a second concept topic. This concept topic became “Semantics in HTML.” This topic describes how generic containers (or “semantically meaningless” elements) are different from other HTML elements. Some users may not need to understand why generic containers function the way they do in order to use <div> and <span>. My hope is that the task topics are sufficiently independent of the concept topics to allow these users to follow the directions and still use the generic containers properly. Users who wish to know more about generic containers, or have questions based on the task topics, should find their most important answers in the concept topics.

Both concept topics reside at the same level as the supertask in the topic model hierarchy. I struggled with this choice—I feel like the concept topics could conceivably be child topics to the supertask. However, I made my ditamap so that it could theoretically act as a child ditamap in a larger topic model about HTML and CSS. The concept topics I’ve created for this generic container ditamap could be reusable content elsewhere in the general HTML and CSS ditamap. Users looking at a manual on generic containers in HTML may even be seeking concept information on semantics in HTML—I didn’t want to bury the concept topics under the supertask or confuse users looking at the supertask. I feel this topic model structure best supports my purpose of bettering user understanding of HTML, CSS, and generic containers by offering concept information up front. That being said, I think there is an argument for structuring the concept topics as child topics to the supertask, as both concept and reference topics are meant to support and supplement user tasks.

Finally, I created two reference topics. One reference topic, “Generic Container Comparison & Examples,” acts as a quick reference guide for users who want a side-by-side comparison of <div> and <span>. I felt that this reference topic best supported the “Choosing a Generic Container” supertask. This entire reference topic supplemented the user’s choice—it’s also possible that, by looking at the reference topic, some users may bypass the child task topics entirely. Users who are confused after reading the <span> or <div> task topics could also return to the supertask page, see the comparison reference, and realize the differences between <span> or <div> through example instead of detail-heavy paragraph text.

The other reference topic is “HTML & CSS Term Glossary.” Like the comparison of generic containers, this glossary includes examples and definitions. However, this glossary collects terms from multiple task and concept topics and can service users who are approaching the manual from either of the topic concepts. If I were to repeat this project, I would try to use the “copy-to” attribute to make the glossary reference topic a child of both concept topics.

Overall, I feel like I created useful, modular user-support content. The topic model architecture supports my primary goals of encouraging proper use of HTML’s generic containers, <span> and <div>, and developing user knowledge of HTML and CSS as a whole.