To: Chris Lindgren

From: Cindy Croy

Subject: DITA topic model design rationale

Date: December 23, 2015

What is included

The overall goal of my topic model is to get someone who is totally new to Reveal.js started with three basic types of slides needed for a presentation. These are a title slide, a bullet list slide, and a slide with an image. The first topic in my map is a concept topic because I believe giving the users an overview of the concepts involved will make it easier to understand the reason for doing certain tasks. The next topic is a task topic that describes what a user needs to do to get the demonstration HTML file ready for creating a three slide presentation. I chose the demonstration HTML file as a starting point because it provides the information for all the features provided by Reveal.js, and I believe it is easier to modify files than it is to create new ones. However, it needs to be simplified for a three slide presentation, and the “getting started” task gives steps for the simplification.

The next topic is a concept topic. It started as a task topic with a simple list of the remaining tasks, but having a list without more information wasn’t very useful or informative. As a result, I created an empty concept topic that would only show the short description as an introduction to the task topics. However, since the override task was in there as well, this needed a little explanation in case a user landed on this page first. The next topics are the basis for the three slide presentation. The first of the four is a concept topic that gives tips for choosing which method to use for overriding the Reveal.js defaults and serves as an introduction to the three tasks for accomplishing the override. Again, this one started out as a task topic with only a list of the subsequent tasks. The next three topics include the steps for making each of three particular types of slides. With an intro slide, an image slide, and a bullet list slide, the user will have a basic template for making a full presentation. Additional features from the demonstration can be added later as the user gains experience.

Design decisions

As I started to write this memo, it occurred to me that a user who is fairly naïve about HTML and CSS (like I was when I started using Reveal.js) might find it helpful to have a task describing where to find the relevant files and how to prepare the files for use in a presentation. For this reason, I added the “Getting started” topic.

Originally I had the first concept topic at the bottom of the page. However, I believe it makes more sense at the beginning since it will give users an overview of the concepts involved which should make it easier to follow what is happening in the tasks.

The overriding Reveal.js tasks provide the reusable content because they contain options for overriding the defaults in the Reveal.js demonstration. Since the user may start with any of the three slides, instructions for resetting these defaults will be necessary with all three of the slide topics.

I could not make the elegant ways of making the “resetting defaults” task reusable work for me, so I resorted to simply making it a child topic of several other topics. Bellamy suggests that the best way to reuse a topic in this way is to use the “copy-to” attribute, but I couldn’t find a clear description of how to use the attribute. I do not really like the way using child topic method looks in the pdf output – I don’t think it needs to reappear with each of the “making slides” instructions. If I had more time, I would figure out how to change this so it wouldn’t output this way for pdf’s.

Originally, I made the reset topics children of an “override” task, but the task was only a list of the different possible ways of resetting the defaults. Having the parent topic lacking in any useful information seemed pointless, so I decided to make the parent topic a concept topic with tips for choosing a particular reset method. Similarly, the “create slides” concept topic also started out as a topic task with child tasks just listed. Again, this seemed pointless, so I made an empty concept topic with a just a short description to introduce what was happening with those topics. However, the “overriding” topic seemed to need a little more description since there is no slide created from following this topic. Therefore, I added some text to the concept topic that describes the purpose of the overriding task.

I’m not sure if using concept and task topics in this way represents a “best” practice – or even a good one – but it seemed the best way to handle the situation.

When I originally wrote the task topics for resetting the Reveal.js defaults, I was using line numbers in the Reveal.js CSS file to refer users to places where they could make changes. Initially I was thinking it wouldn’t matter how others style their code because I was referring to a specific file that came with the download and that this would be unlikely to change. Then I thought that there may be Reveal.js updates that involve modifications to the demonstration slides, so this file may change in later updates so maybe line numbers wouldn’t work so well after all. Therefore, I modified all the task topics so they didn’t have line numbers.

If I had more time would be to set up the HTML output such that there would be a link back to the main title page which shows all the links. The only way to get out of the “Important Concepts”, “Getting Started”, “Create Slides”, and “Resources” links is to use the back button on the browser. As a user I would prefer if there was a more direct way to go to the next topic. Adding such a link does not contribute much to the user accomplishing the goal of making a three slide presentation, but it would contribute to the usability of the site. The page feels incomplete if there’s nothing on it except text.