The Center for Information-Development Management
710 Kipling Street • Suite 400 • Denver, CO 80215 • 303/232-7586 • www.infomanagementcenter.com

Article Reprint

Don’t Wait: Develop a new Information Model now

JoAnn Hackos, Comtech Services, Inc.

You’re seriously considering a move to XML—and topic-based authoring.

You’re getting ready for a move to the DITA standard.

Implementing a Component Content Management System is your major goal for 2010.

If you agree with any or all of these statements, it’s the right time to develop a new Information Model for the information you prepare for customers. But just what do we mean by an Information Model?

- What is an Information Model?
- Why is it so important?
- What if you already have a well-defined organization for your FrameMaker books and chapters or your Help files?
- Why won’t your existing architecture work for XML and DITA?

Here is Don Rohmer’s assessment of the importance of an Information Model:

> “To describe changes that have made my life easier, I wouldn’t know where to start, so I would like to mention the exciting potential of information modeling. A major problem of our day is that everyone is drowning in a flood of information, much of which is not relevant to their needs. Widespread adoption of information modeling techniques could provide everyone with a map through the swamp.”

Note: Don is a Senior Technical Editor at Citrix Systems

Information Model Defined

Richard Saul Wurman originally defined information architecture as the act of designing an information model:

> “I mean architect as in the creating of systemic, structural, and orderly principles to make something work—the thoughtful making of either artifact, or idea, or policy that informs because it is clear[1].”

An Information Model is a set of principles that define how you intend to structure the information you develop. For example, you are developing information that will help customers use your product efficiently and effectively. You may state in your Information Model that the customer is best supported by learning to perform step-by-step procedures. For that purpose, you will define the purpose of the task information type, for whom it is intended, and its basic structure.

If your customers are best served by thorough descriptions of commands in a programming language, you will define a command reference information type. As you identify the information resources your customers require, you will define the basic set of information types that will meet their needs.

Then, for each information type, you must define the constituent parts, their purpose, and what content they will contain. For example, your command-reference information type may name the command, state its purpose, provide its underlying structure, and include examples of its use.

You may even go farther than the core information types and define larger structures, such as document types, that specify the standard structure of a user guide, a product specification, or a quick reference card.

Information architecture is the activity, and the Information Model is the result.

Why is an Information Model so important?

You may be wondering why you have not already defined an Information Model for the content you produce. In some cases, you may have the rudiments of an Information Model already in place. In organizations that produce structured, topic-based help systems, we often see a fairly well-defined Information Model.

One organization, for example, defined three base information types for their help system and used tabs to differentiate among them in final output. They defined the concept, the procedure, and the reference topic. A training organization, using the base Information Model provided by instructional designers like Ruth Clark and Robert Mayer may have decided to include facts, procedures, concepts, processes, and principles to form the core content of their training courses. Others may be advocates of the seven information types of Information Typing™.

In each case, the careful definition of the purpose and structure of the information and document types allowed writers to produce consistent content that could be intertwined in various deliverables. The Information Model provided a set of rules that

[1] *Information Architects*. Graphics Press, 1995.

the writers followed to ensure consistency, even when the outcome included the work of more than one individual.

However, in many organizations, the lack of a well-defined Information Model results in various idiosyncratic styles and content. Each writer, working alone to produce a unique deliverable, works differently and produces different results. The result is content that is often poorly defined and rarely interchangeable with the work of other writers.

Why is an Information Model needed for DITA?

Not only does DITA require that writers develop standalone topics that are interchangeable with other topics written by different writers, it requires a degree of consistency in structuring and labeling content that is rare in a desktop publishing environment.

Developing DITA topics that foster a reuse strategy requires a collaborative work environment and an agreement to follow a comprehensive set of rules and requirements. The Information Model instantiates the set of rules and requirements that everyone agrees to follow.

In addition, however, DITA requires consistency at the element level. Take, for instance, the DITA strict task model. This model requires that writers create a series of step-by-step procedures using the <step> element, followed by the <cmd> element to ensure that each steps begins with a command statement.

Writers must develop their procedures in a DITA task using <step> and <cmd>. An Information Model would reiterate that requirement, ensuring that everyone understands the importance of beginning a step with a command. If writers want to include substeps as part of a step, they must use the <substep> element and another <cmd>.

Sounds perfectly logical and should be stayed with an example in the Information Model.

However, what happens when someone decides they want to add substeps without writing any text for the first-level step? What's to keep someone from creating something that seems "clever" to them, is allowed in DITA, but makes no sense to a reader?

```
<steps>
    <step><cmd>NO CONTENT HERE</cmd>
       <substeps>
             <substep><cmd>Turn on the engine.</cmd></
substep>
             <substep><cmd>Do something else.</cmd></
substep>
       …
      </substeps>
   </step>
 </steps>
```

The solution must be stated in the Information Model—substeps are not allowed if there is no step. Of course, if the style sheet expects content in the step command, it will produce a number for the step but no text. The reader will think that something has been omitted.

The same is true for other possible examples of "clever" XML markup or simply to maintain "iron-clad consistency" in the way that all the writers mark their content with XML tags. Because style sheets and other processing depend on consistent DITA tagging, the rules for tagging must be included in the Information Model. Without the rules, writers may introduce tags that are not part of the style sheet or create errors in processing. The Information Model builds consistency in style and XML mark up.

We have seen organizations that didn't think they needed an Information Model. Every writer was free to use DITA any way they wanted. The inconsistencies in presentation caused chaos, limiting the potential for reuse and introducing processing errors that required many hours to track down and correct.

Certainly, organizations have built comprehensive Information Models before DITA. Ginny Redish and I helped build one for Hewlett-Packard in the early 1990s. But XML introduces the need for another layer of rules that create consistent labeling of content using semantic (steps, substeps, commands) XML markup rather than the formatting-related markup (paragraph, numbered list, bulleted list) of desktop publishing.

Information Models are critical to DITA success

Not only does your Information Model define the rules of engagement with your customers and your content, it also provides an opportunity for your organization to carefully define the strategies you will use to enhance the reuse of topics. One of the most important cost savings that DITA enables comes from reuse, either at the topic or element level. But that reuse does not come without careful planning.

DITA provides you with a number of tools to enable reuse from the basic href to combine text and graphics through the mechanisms of content reference and, with DITA 1.2, key reference. In addition, you can achieve successful reuse by modifying style sheets, using conditional text variations, integrating with databases, and so on.

By studying the options and understanding the opportunities, your organization can achieve significant levels of content reuse across deliverables. However, these strategies must be defined in your Information Model and processes outlined that ensure that the opportunities are recognized and acted upon.

Organizations should view a move to DITA as an opportunity to rethink its approach to content. As Don Rohmer suggests, we have long produced too much content that no one uses. The Information Model is the outcome of the rethinking process.

Too often, people view a move to DITA as "just another tool." Rather than using MSWord or FrameMaker, writers are now using an XML editor. The work practices remain the same, and the legacy content is migrated. As a result, a great opportunity is lost.

With a new, thoughtful, and comprehensive Information Model you have a great opportunity, perhaps the only one, to rethink the content and do something truly innovative and exciting. Don't wait: Now is the time to begin your Information Model.