

## 1. EVEN THE MOST COMPLEX DEVICE IS ULTIMATELY A COLLECTION OF BASIC PARTS

*It's the same for complex technical documentation*

As digital information experts we often get deep into the arcane details with our terminology and explanations. And it sometimes makes it hard to pull back to seeing things the way they look to the outside world. Linguist Steven Pinker calls it 'the curse of knowledge'. Things are no different when it comes to a concept like 'structured content'.

*The Basic Concepts of Content Management*

Understanding the way Component Content Management Systems (CCMSs) and Product Information Management Systems (PIMS) organize content, and why, requires a paradigm shift for both managers and content creators, a shift that can seem somewhat counterintuitive. Essentially, these systems organize content by its type rather than as longer documents composed of a single file. The common types are:

- Concept (generally a text that conveys meaning or context)
- Task (an ordered list of steps to accomplish something)
- Reference (information like specifications or a parts list)

These topics are created individually in the system which then tags them with attributes. To create a manual, for example, the content creator uses a taxonomy map to organize them into the manual. This may involve connecting that taxonomy map to a series of files like a Safety Warning, a Legal Disclaimer, an Introductory Concept block of text, an installation Task List, a parts diagram (Reference type), a Task List for operating the product, etc. Each of these files becomes a component that can be sliced and diced, moving pieces around, editing or updating individual sections, and more. When they are complete the manual can be published to virtually any digital media format.

*A Simpler Way To Look At It*

Did you see how quickly that got more and more complex? Let's consider a kind of simple metaphor. Every physical product, no matter how complex, can be broken down into static parts, at its most basic level. Knobs, castings, chips, wire, etc. If you took all those parts and dumped them out of a box unassembled you might not have any idea what they do or how they go together. But if you have plans and assembly instructions you can turn all those static things into a working device.

Taxonomies are the plans and the assembly instructions for digital content. The 'parts' in the box are those pieces of content mentioned above. It is very possible, even likely, that many of those parts have other uses, in other products or other plans. When your content is organized like this, that ability to reuse it, without copying from other documents, is a critical aspect of information design. It saves enormous amounts of time in any number of ways. You might have a Legal Disclaimer that gets referenced by dozens of taxonomy maps for use in dozens of disparate documents, without copying anything- it is just referenced by the system and sent out whenever and wherever it is published (without affecting any of the other docs utilizing it). And if a change is required that affects all instances of the legal disclaimer, it is changed once in the system, and everywhere it gets used in a document will reflect the updated version.

*The Tools and Processes Have Evolved*

These systems offer powerful ways to leverage content that were simply too labor intensive in the past. There is a learning curve, and implementation requires planning and configuration, training and content migration. Dakota can help you every step of the way from system design and recommending solutions, to training and working through the migration process. Yes, there is complexity, but once your team has made the switch, you'll view your valuable content in a whole new way.

*Talk To Us Before You Make Decisions*

Facing a content management or product information overload? Give us a call and we can talk you through your options. It is especially critical to have that conversation before you choose a solution— there are many options and we can help you tailor them to your specific needs before you commit.

