



Why CCM is Not a CMS: Or Why You Shouldn't Confuse a Whale with a Fish



Howard Schwartz, SDL Structured Content Technologies

People are beginning to realize it is a category mistake to call some kinds of systems a “CMS” (Content Management System), when what they really are referring to is a “CCM” (Component Content Management) system. Under certain circumstances, the difference is critically important. By making this category mistake and confusing a “CCM” system with a “CMS,” some organizations are failing to convince their management that they need a specialized system called a CCM. Their management or IT organizations think they already have one, when in fact they do not. Organizations that succeed in adopting a CCM system have argued persuasively to management (if not to their IT organization) that a CCM system is not a CMS. Just as a source control system (which does versioning) is not a CMS and just as a Web Content Management (WCM) system is a specialized type of content management system, so a CCM system is a unique specialized system for the management of another critical enterprise function: namely the management of technical information.

To understand this question of categorization, it is perhaps helpful to look at a less technical analogy. At first blush, it would seem reasonable to classify a whale as a fish. After all, a fish is a sea creature that swims and lives in the water. And by these criteria, a whale certainly would fall into the category of a fish. But we tend to reject this categorization because our scientific classifications trump the more intuitive categorizations. A whale, say scientists, is really a mammal because like other mammals, whales breathe air into their lungs, are warm-blooded, and the females feed their young milk from mammary glands. A whale is a mammal and not a fish; therefore, we give scientific classification priority on how we organize our thinking.

This analogy is useful when we decide what kinds of systems to include in the category of CMS (“our fish”). It is not the case that the systems we are calling a Component Content Management (“the whale”) system can't fit into that CMS categorization. They can. But that categorization misses something essentially important about these specialized systems when they are lumped into the undifferentiated category of “CMS.” We shall say more about this below, but essentially a CCM system has capabilities that are not in every old CMS. A whale could be a fish but....

THE CONTENT MANAGEMENT EVOLUTION

This kind of category mistake is common when there is a para-

digim shift occurring in technology adoption and a new technology category is emerging. Indeed, we have seen this before in the emergence of the content management category itself. Remember the days long ago (the early 90s!) when we had only “Document Management” systems? When the content management category first emerged no one had heard of that category either and the early CMS systems had to differentiate themselves from Document Management systems. Then HTML and the Internet came along and imposed new expectations on corporations for managing their interaction with customers over the Web. With this change emerged the Web Content Management system. That category did not exist until the internet had redefined how organizations needed to interact with their customers and prospects. Now it is almost a de facto standard that every marketing organization has a WCM system. A WCM system differs from other kinds of content management systems because it is focused on a particular problem and process in the enterprise (managing web sites) that have distinctive challenges and business objectives.

The same paradigm shift is now happening with the emergence of CCM as a system distinctive from a CMS. This emergence is part of a much broader shift that is occurring in how organizations manage technical and other business-related information across the enterprise. The shift occurring in the management of technical information is arguably as momentous as the impact that the internet initially had on corporate marketing organizations in the mid-90s. What is driving this transformation for technical information management is the shift from a book-oriented writing paradigm to topic-based writing using the XML standard called DITA (Darwin Information Typing Architecture).

The manifestation of this shift is the changing role of traditional technical writers or authors into true “information developers,” a term that has been around for a while but not truly realized in the publications world until now. Now publications development is becoming much more like software development. In this new world, many individuals contribute “modules” or “topics” in their own particular area of expertise. They create these topics “disembodied” or “contextless” and then roll these topics into larger blocks of content that ultimately culminate in publications or content delivered to customers. The implications of this shift are enormous for organizations that have crossed this chasm from “book methodology” into “topic-based authoring.”

One such implication is the need for technology infrastructure of a sort that was not previously needed. It is in this context that the emergence and importance of CCM becomes understandable and critical. In the past, when working in a book paradigm, technical publications organizations only had to version whole documents or chapters, for the most part. While they did seek to reuse content across publications, they typically did so only by cutting and pasting from book to book or chapter to chapter. Since books or chapters were the unit of tracking, versioning was important but could be handled with naming conventions on the files system or with a source control system.

By contrast, with the move to DITA and topic-based authoring, what once was a suite of 200 documents for one of our clients has become 70,000 topics that all have to be managed and versioned. For other clients with a larger suite of documents, the number of topics can reach into the hundreds of thousands or even millions. These topics, moreover, may need to be managed in multiple languages (70,000 times the number of languages). To complicate matters further, these topics can be used across many different deliverables. The same topic may have to appear in an administrative guide and user guide. It may also show up in web pages, PDF, Help files, and other outputs. It may have variations for advanced and new users. Tracking the version of each topic and graphic that goes into each deliverable is simply not possible when managing on a file system, especially if you have documentation of any substance or a requirement to manage versions and their relationships to each other and to larger publications.

CONFUSION BETWEEN THE CMS AND CCM CATEGORIES

This brings us to the current confusion over the content management system. When many organizations begin to understand DITA, they ask themselves, “Do we need a CMS?” (See our earlier white paper, “Dare I Do DITA Without A CMS,” available at <http://www.sdlxyssoft.com/en/knowledge-center/white-papers/>). After trying DITA on a file system in some small prototypical way, organizations of any size quickly realize that they will need a process to manage the tens of thousands of topics that they have. They will also need a way to track which versions go into which deliverables or releases. In addition, they will have to track all the various versions of their graphics and screenshots and on top of that they will also have to track all the language variants of everything.

“Aha,” they say to themselves, “We need a CMS.” But here is when they are in danger of making a significant category mistake. What they really need is a CCM system—not a CMS. What’s the difference? There is a whale of a difference.

Forget for a moment the names CMS and CCM—the names don’t really matter. What an organization needs to manage DITA is a system that can track not only versions of topics and graphics but relationships among topics, graphics, maps, publications, and deliverables. In DITA, a map may refer to five hundred topics. That map may itself go through dozens of versions, and the

topics themselves may each be versioning independently on their own. The topics in turn may both refer to graphics that are going through versions and to components of other topics (conrefs) which are themselves going through versions.

The problem of managing DITA is not the versioning of discrete “objects,” it is the problem of versioning all the objects in relationship to each other at the same time. It is tracking the moving relationships of all these components to each other and not the versions themselves that presents the problem of managing information in a topic-based architecture. Without “understanding” the particular language of DITA (this flavor of XML), the tracking system can’t possibly understand or manage the relationship of all the moving parts. It is this linking of versions to each other, to larger blocks, and still larger blocks into publications that the standard run of the mill content management system can not manage out-of-the-box. The name Component Content Management is now being adopted by analysts and users to describe a system which has this key layer of functionality that sits above and beyond the standard versioning and workflow of every standard content management system. This layer of logic is at the heart of what the CCM system delivers, and it is at the heart of what makes DITA deliver an ROI.

The need for complex versioning among topics and parts of topics is the reason that analysts and vendors are starting to differentiate systems that have versioning from a system that is specifically tailored to understand DITA. This concept is hard for people to initially grasp precisely because we are in a paradigm shift. Most everyone is familiar now with a CMS, but only some people have heard of CCM and DITA. The risk of not differentiating the CCM system from a CMS is that the rest of the organization may already think they have one when, in fact, they don’t. And when your organization wants to adopt one, your management or IT organization may come up with the infamous “We have one already.”

A funny but illuminating instance of this problem arose at FICO, one of our large Bay Area software clients, that was looking seriously at DITA and realized they would need something to manage the linking and versioning. At first the conversation went something like, “We need a CMS.” When other departments heard they were thinking about a CMS, those other departments said “We need one too,” and started to raise a fuss that they wanted to add in their requirements. Ann Kana, Vice President of Product Services, grasped the difference between a run-of-the-mill CMS and one specialized for DITA and guided her group to call the system a “DITA Management System” as they went for budget. By doing so, the organization was able to differentiate their requirements and needs from those of marketing and legal, and they had an answer when IT said “We already have one.” By deftly categorizing what they needed, they were able to drive toward a purchase that met their requirements. By contrast, we have seen numerous organizations that have become bogged down in discussions with IT on why they can’t use the already

purchased enterprise CMS. IT can't understand DITA easily and therefore can't understand why the regular versioning of a CMS won't work. "We already have a corporate standard—why can't you just use that." The problem is that the corporate standard is not specialized for DITA. It is an off-the-shelf versioning or collaboration platform. But that is not enough for the management of DITA. Since it does not understand DITA and cannot track the relationships of objects as they version in relationship to each other, it won't deliver the ROI that DITA promises.

We believe that the emergence of the CCM category is just beginning. Many organizations are deploying DITA and recognizing that they need more than just versioning to manage the complexity of how topics roll into maps and maps roll into publications. They need a system that prevents links from breaking when topics, graphics, and conrefs are versioned. They need a system that allows them to more easily manage the relationships among maps, topics, and publications. CCM is the latest evolution in the specialization of content management. And just as a source

control system (which does versioning) is not a CMS and just as a Web Content Management system is a unique system, so is CCM a unique specialized system that is essential for the management of a critical enterprise function: namely technical information.

What is ultimately at stake for those in Technical Publications, now "Information Development," is to realize that "CMS" is a category that hurts them more than it helps. If they ask for one, they will likely find that the IT organization has already invested in one. And chances are that that CMS or "Collaboration Platform" is not one that can understand DITA natively. It may be able to check in and check out topics and versions. But those who understand DITA know that versioning is not the key issue. The key issue is managing the relationship of versions to each other. And to do that, the CCM system must understand DITA. It must understand what a topicref, conref, and ditamap are, to name only a few examples. This is why CCM is not a CMS and why a whale is not a fish. 🐋