

The Parts and the Whole:



shutterstock.com/file404

By FABRICE LACROIX

Publishing Technical Documentation



TODAY, SEVERAL CENTURIES after the invention of the printing press, we can legitimately ask ourselves whether there's any value to preparing technical documentation in book form. Isn't a printed book—a *whole*—antithetical to the notion of component-based authoring, in which each component is a free-standing *part*? In this article we examine how the book, with its fixed structure, is affecting the way we access and use information. Then we look at some alternative approaches to publishing technical documentation. These reflections stem from our in-house experience and the difficulties we encountered at Antidot upon switching to structured writing.

Antidot is a software vendor that specializes in data management: search engines, text mining, and semantic enrichment. Our mission is to help our clients create value with their content. Five years ago, we embarked on a top-to-bottom redesign of our technological platform. We threw away two million lines of code in order to start with a blank page. This was a golden opportunity to create a new documentation process from scratch.

We decided to go with structured documentation: we hired a technical writer, we selected and purchased an authoring tool, we set up a process, etc. And it works! Everything you've heard about the benefits of component-based writing really is true—it's not just a marketing pitch! We write faster and at a lower cost, we reuse content, and maintenance is easier—just as advertised. It worked so well that very soon we had progressed from a few hundred to a few thousand topics, and from just a handful of guides and manuals to more than a hundred. We were proud of the result. Never before had we written and documented so much, so well.

One day we asked our customers and partners for their feedback about our documentation. In our minds, it was less about asking an open question than about giving them an opportunity to congratulate us and tell us how impressed they were by the quality and quantity of this new material—a perfect counterpart to the richness and beauty of our products. However, the replies were very disappointing. We did not receive praise but complaints: “Too complicated. We can't find anything. We don't know where

to look.” Our in-house consultants and support staff were saying the same thing. What's more, they told us things were getting worse. The more documentation we were producing, the more they would complain. In addition, the support burden began to increase, as did the problems involved in transferring knowledge.

How could this have happened? At Antidot we are producing software designed to facilitate access to information, with the most advanced companies calling on us to help them make optimal use of their data and transform their businesses. How could we fail with our own documentation? What a shock that was for us! We had to figure out where, when, and how it was so. We conducted a thorough analysis to understand where the problem was.

The content we were producing wasn't the issue, although naturally there's always room for improvement. Nor was it the authoring tool. Nor even the process. In fact, our mistake was rooted in the very last step: publication. We would routinely press a “publication” button and our authoring tool would generate dozens of PDFs, HTML pages, or ePubs. We realized that we were locked into an age-old mindset that goes back to one of humanity's greatest inventions: the printing press. We were *still* printing. Because even in virtual format, generating digital books is printing. And what we were basically offering our users with those books was content—nicely organized content, for sure, but content that imposed a fixed, linear path on the reader.

At this point, it is essential that we differentiate between the book as a structure as opposed to its content, or in other words, the items of information that it conveys. As an end object, a book is a story told by the author, a charted path to be travelled, a set route for taking us from point A to point B. Although that kind of journey is the essence of literature, technical documentation is a different kind of experience, one in which an exhaustive, linear reading makes no sense. In technical documentation, a book is rather a practical means of grouping information. For that matter, the table of contents, the index and bibliography, the search engines provided—they're all there as an excuse, to serve as back doors through which users can escape

the book and its fixed path, to let them access a random piece of contents, to seize the information they need and leave. As for the notion of reusing content, the cornerstone of component-based writing, it is also a means of getting around this rigidity inherent in the book. A means of telling several stories using the same elements—of creating different paths. But then what good is it to publish books? Do we still need to assemble topics into published works? That’s a completely different argument, one that has its proponents and its detractors. For my part, I still believe the answer is yes, because a book also provides a context. It reflects the logic behind the product and how it is used. Just as the topic’s content is one piece of information, the book’s structure offers another one.

At the same time that we are capitalizing on the concept of component-based authoring based on topics assembled into books, we have to change the publication step. In the current approach, topics are just a means of more efficient production; the overall objective is still a book. Therefore users need to search within books, download books, and read books. We need to reverse that paradigm: the user should be searching within topics, reading topics, and navigating from topic to topic, whether linearly by following a predetermined reading path known as a book, or by following links and other ordering methods that can then be constructed dynamically.

For this, we simply need to stop printing! We need to provide a form of search and navigation in which the key that opens the door for users is no longer a book, but a topic. This means that we must keep the topics alive and avoid losing them during the book generation process.

Let’s take a closer look at the impact of switching from the book to the topic as the primary editorial unit for publication.

Publishing Books

First, let’s examine the dominant practice: the impact of the book as the unit of publishing at the heart of the editorial process (see Figure 1). What we find is a widely

recognized paradigm that is deeply rooted in library practices. Writers communicate with users in terms of published work: books arranged on shelves. The essential metadata are those attached to the book, because they serve as access keys: user manual, maintenance guide, for beginners or experts, free or reserved access, for version such-and-such of the product, and so on. But the result in terms of accessibility is dramatic.

When searching for information, two steps are involved. First, you have to know which book you need, and second, once you’ve identified and downloaded the book, you need to navigate through its contents using the table of contents or the index. Scholarly and cumbersome, isn’t it? Moreover, in order to identify the appropriate book for your needs, you either need to use a book classification system built from metadata—a tedious and slightly outmoded method in the digital era—or you need a search engine.

And that’s precisely where the limitations of the book-based paradigm are most apparent. The search engine is designed to meet the needs of users who don’t know their way around, such as novices who aren’t familiar with the documentation available. They type in a few keywords and in response they get a long list of books. That’s the first step in a frustrating journey, because in most cases they’re not looking for a book—they’re looking for information. They’re forced to open up each book on the list one by one, download it, browse through it, and determine whether the book actually corresponds to their specific problem and contains the right information. But is it the best information on the subject? Is there any other information available? In order to be sure, they’ll need to download, open, and read every book listed in the results page. It’s a tedious, frustrating, time-consuming, and absurd process. As for the search engine itself, how much relevance can it offer when it’s indexing documents that extend to dozens or hundreds of pages? Given their size and content, some large, generic books will always be part of the results, regardless of the query. The user experience is weak. All of the analytical tools that have been created

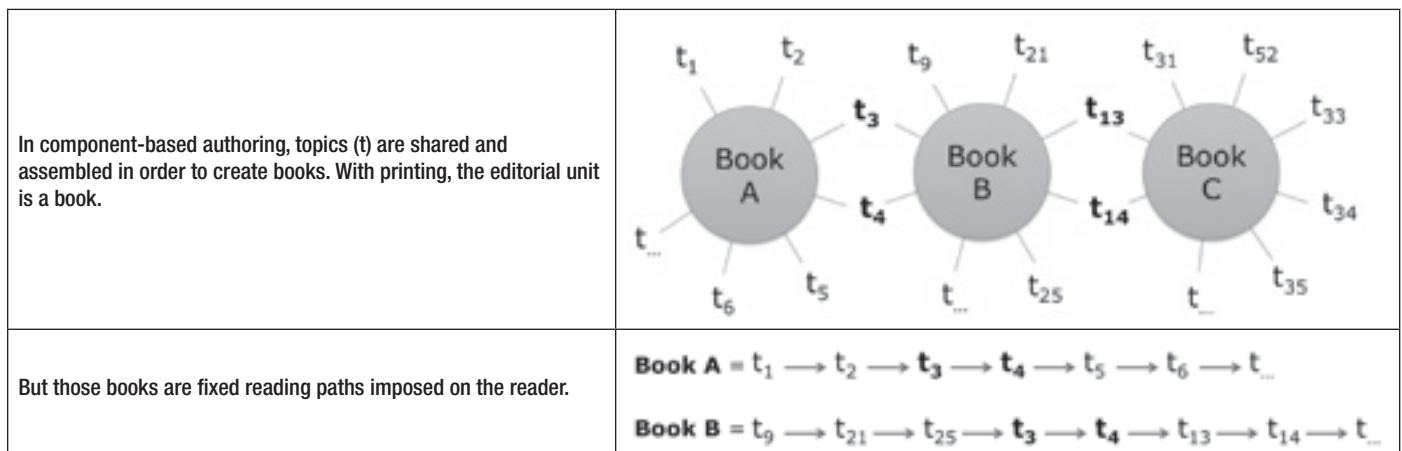


Figure 1. Component-based authoring in book format

in order to understand what users read are also biased and therefore useless, because user behavior is itself fundamentally distorted by that forced process of downloading and hunting for information.



Figure 2. Searching in books. Query: “filtering replies”

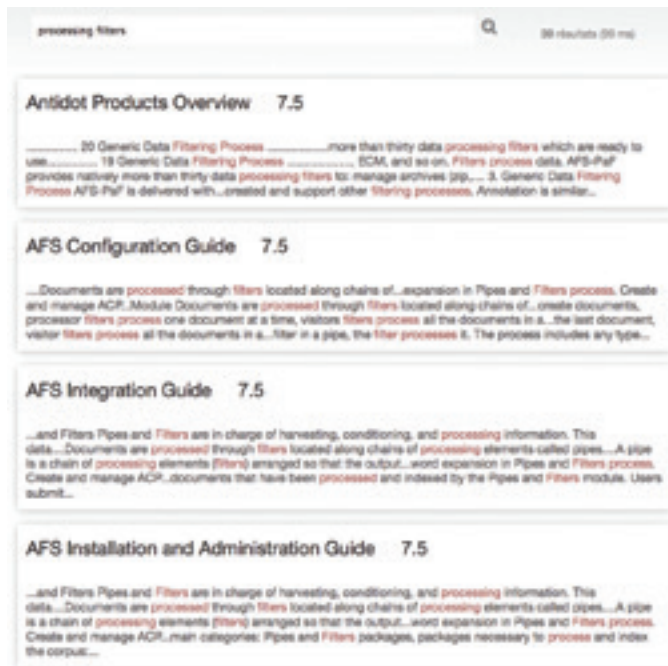


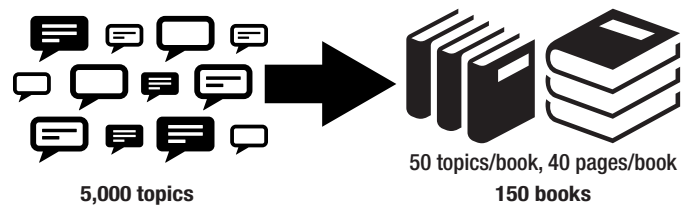
Figure 3. Searching in books. Query: “processing filters”

The searches “filtered replies” (Figure 2) and “processing filters” (Figure 3) return almost the same content and are just like most searches: their results always consist of largely the same list of books, so they don’t offer a very useful service.

Publishing Topics

Now let’s look at what happens when the topic is the core element of not just the authoring process, but the publishing process as well. What does it mean to “publish a topic”? It means that each topic is exposed as is, in its raw format (in most cases XML). This way, structural information isn’t lost or flattened out, as happens during the printing process (i.e., generating PDFs or HTML).

The first consequence is that the search engine is able to index each topic individually, making maximum use of the topic’s structure and metadata. Thus, when the user submits a query, the search engine responds with topics, and since only relevant topics are shown, the results are vastly more useful. Let’s consider the example of a corpus of 5,000 topics that make up 150 books, with an average size of 50 topics and 40 pages per book.



In a book-level search (the usual practice), the search engine will easily return more than 50 books. Fifty books x 40 pages = 2,000 pages to browse through.



Whereas when topics are indexed, the search engine will yield, at most, 200 topics—the equivalent of 200 pages of content. That’s 10 times more efficient.



And that’s not including the user’s enhanced ability to apply filters (faceted searches) by making combined use of the topic’s metadata and the metadata of the books in which those topics appear.

With topic-based publication, the search engine generates topics in response to a query. This is both more precise and more informative. When a topic appears in several books, the user can choose the context in which to consult that topic.

The second major consequence of topic-based writing is that the reading phase is made considerably simpler, because each topic is accessed and read directly. Users can choose the specific context (i.e., the specific reading path

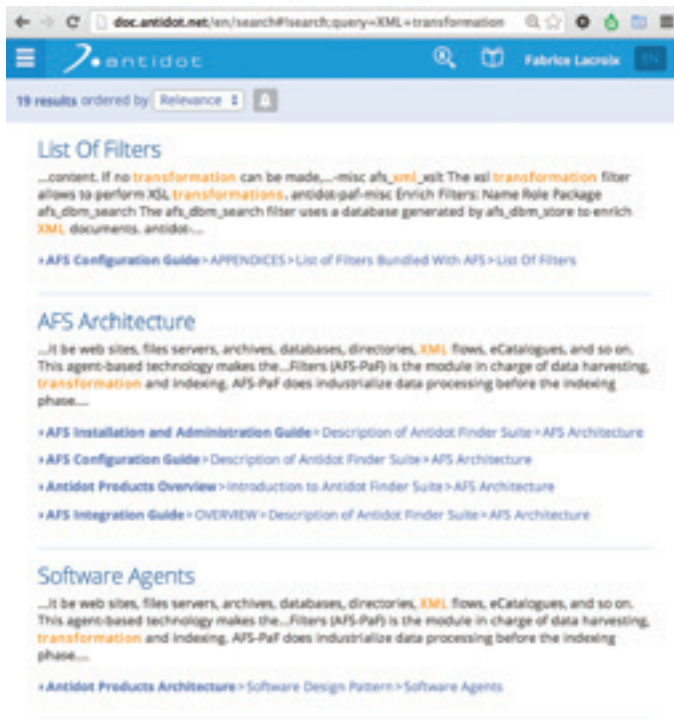


Figure 4. With topic-based publication, searches return fine-grained results at the topic level.

or book) in which they want to read about the topic. They can even maintain their focus on the topic while changing the reading context; in other words, they can read about the topic within one book, and then within another book. It then becomes possible to propose follow-up texts for the user to read or other topics related to the current topic. This can be done dynamically, taking into account both the initial query and the content that users have already consulted: “You’re looking for information on this subject, you’ve already read this and this, we suggest you glance

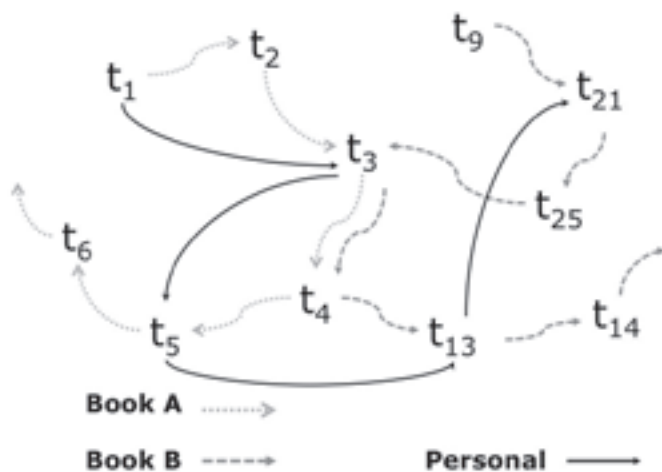


Figure 5. Books are nothing but proposed reading paths. Other paths can be constructed and suggested dynamically.

[T]he use of topics as the basic building block in publishing yields countless ways to enhance features and services available to readers: they can annotate and comment on each topic, create alerts, attach bookmarks, and more.

at this and that as well.” Access and navigation are significantly enhanced and the user experience is transformed.

There are other consequences to topic-based publishing as well:

- ▶ Any analysis of viewed content and user behavior is infinitely more precise, because access to each topic is identified and traced individually, as are the searches that led users to that topic and the reading path(s) by which the topic was consulted.
- ▶ A topic-based approach can be combined with advanced text mining and semantic analysis technology to extract knowledge, to create links and reading paths automatically, and to link topics to third-party content such as knowledge bases or user forums.
- ▶ Lastly, the use of topics as the basic building block in publishing yields countless ways to enhance features and services available to readers: they can annotate and comment on each topic, create alerts, attach bookmarks, and more.

Needless to say, all of this is complex and requires an entirely new type of publishing tool, one that can handle a comprehensive range of issues related to security, access filtering, customization, and support for content variants (conditional text, audience-based content, ditaval file management, etc.). This technology already exists and is ushering in a new era for making maximum use of technical documentation, with a significant and measurable impact on support costs and customer satisfaction.

It truly is possible to shift the paradigm and enter a new century, in order to better serve our users and, by extension, our businesses. We just need to stop printing. **i**

FABRICE LACROIX is the founder and CEO of Antidot and Fluid Topics. His career is intimately linked to the development of the Internet and of the Web. He began as a system developer in the telecom industry, and in 1994 he took part in the creation of the first French ISP as CTO, developing many breakthrough technologies. Convinced that the future is in the data more than in the infrastructure, keen on innovation and entrepreneurship, Fabrice created Antidot in 1999. Fabrice is also a board member of different companies, investors, and organizations where he shares his views on innovation and the evolution of the software industry. Fabrice graduated from ENSIMAG and holds a master in computing from the Imperial College London.