

Writing successfully for everyone

Including accessibility in your documentation



Communicator

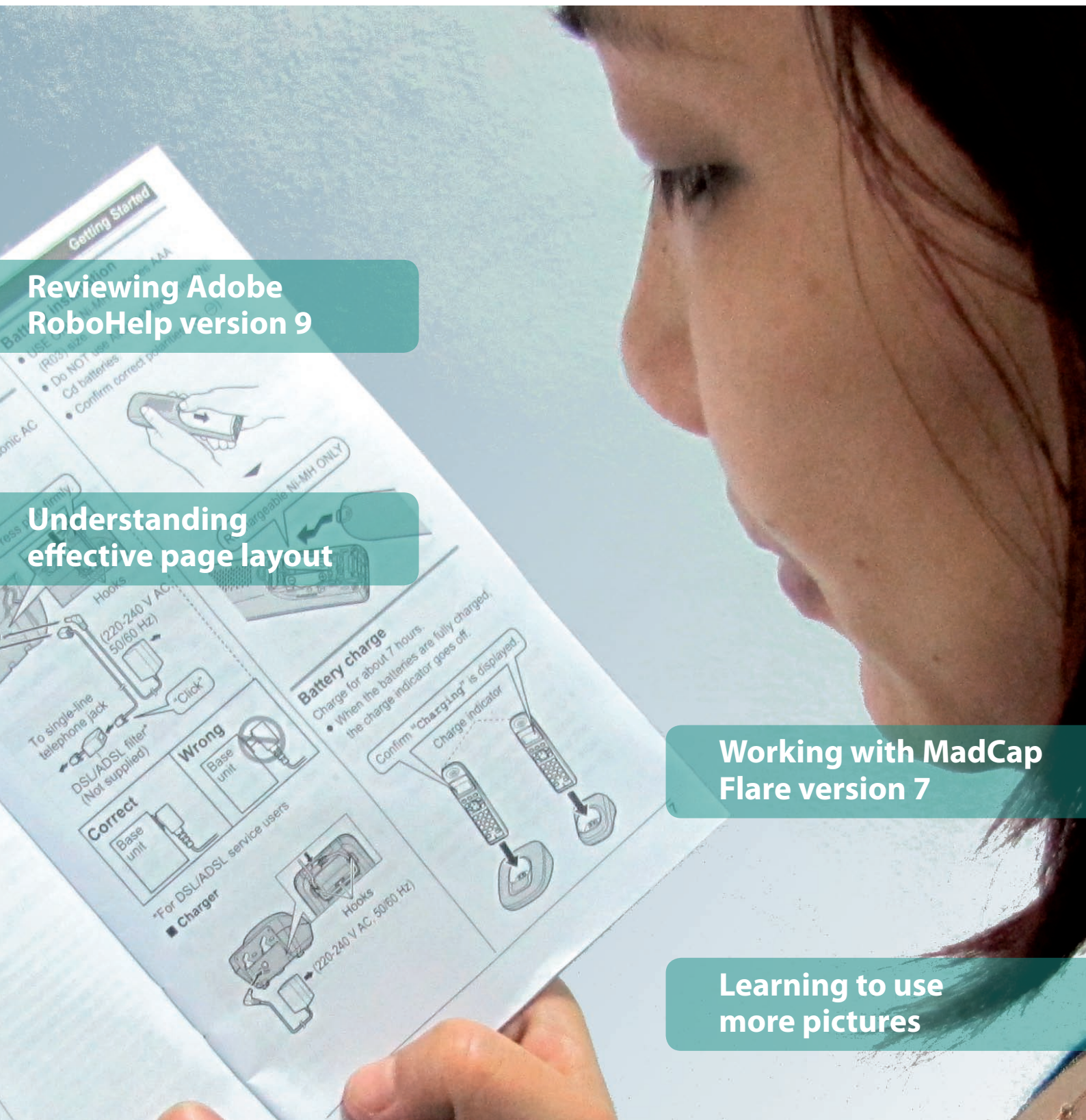
The Institute of Scientific and Technical Communicators
Spring 2011

Reviewing Adobe
RoboHelp version 9

Understanding
effective page layout

Working with MadCap
Flare version 7

Learning to use
more pictures

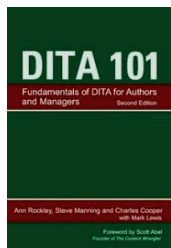


Learning more about DITA

DITA 101: Fundamentals of DITA for Authors and Managers, Second Edition

By Ann Rockley, Steve Manning and Charles Manning, with Mark Lewis

ISBN 978-0-557-69600-0, available at lulu.com as paperback (US\$ 24.95) or PDF (US\$ 9.99), 160pp. Reviewed by Kai Weber.



DITA 101, second edition, by Ann Rockley and others, is one of the best tool-independent books about DITA. For technical communicators and their managers, it's a good primer to learn about DITA in a couple of hours.

Strong context

The book excels in firmly embedding DITA's technologies and workflows in the larger context of structured writing and topic-based authoring. I attribute this to the authors' years of solid experience in these areas, which comes through, especially in the earlier chapters.

'The value of structure in content', the second chapter, illustrates structured writing with the obvious example of cooking recipes. Then it goes on to show you how to deduce a common structure from three different recipes – which I haven't seen done in such a clear and concise way.

'Reuse: Today's best practice,' the third chapter, takes a high-level perspective. First, it acknowledges organisational habits and beliefs that often prevent reuse. Then it presents good business reasons and ROI (return-on-investment) measures that show why reuse makes sense.

Comprehensive, solid coverage

In subsequent chapters, Rockley and her co-authors describe DITA and its elements very well from various angles:

'Topics and maps – the basic building blocks of DITA' expands on the DITA specification with clear comments and helpful examples.

'A day in the life of a DITA author' is very valuable for writers who are part of a DITA project. Writing DITA topics and maps is fundamentally different from writing manuals, and this chapter highlights the essential changes in the authoring workflow.

'Planning for DITA' outlines the elements and roles in a DITA implementation project for the project manager. Don't let the rather brief

discussion fool you: Without analyzing content and reuse opportunities, without a content strategy and without covering all the project roles, you expose your DITA project to unnecessary risk.

'Calculating ROI for your DITA project' has been added for the second edition. It's by co-author Mark Lewis, based on his earlier white papers: 'DITA Metrics: Cost Metrics' and 'DITA Metrics: Similarities and Savings for Conrefs and Translation'. It expands on the ROI discussion of the third chapter.

'Metadata' first introduces the topic and its benefits in general and at length. Then it describes the types and usefulness of metadata in DITA. This might seem a little pedestrian, but it's actually helpful for more conventional writers and for managers. It ensures they fully understand this part of DITA, which drives much of its efficiencies and workflows.

'DITA and technology' explains elements and features to consider when you select a DITA tool, content management system or publishing system. This is always tricky to do in a book as much depends on your processes, organisation and budget. While the chapter cannot be a substitute for good consulting, it manages to point out what you get yourself into and what to look out for.

'The advanced stuff' and 'What's new in DITA 1.2' continue the helpful elucidation of the DITA specification, with comments and examples, that was begun in the 'Topics and maps' chapter.

Mediocre organisation, formal errors

For all its useful contents, the book deserves better, clearer organisation! Redundancies and minor inconsistencies occur as concepts are defined and discussed in several places. For example, topics are defined on pages 4, 24 and 46. While the newly added ROI chapter complements the ROI section in Chapter 3, neither has cross-references to the other. The index doesn't always help you to connect all the occurrences and navigate the text.

Chapters are not numbered, yet

numbering of figures in each chapter starts at 1. This is not a big problem, because references to figures always refer to the 'nearest' number; it's just irritating.

The book contains several errors, which add to the impression of poor production values. They don't hurt the overall message or comprehensibility, but they are annoying anyway. The illustrations on pages 72 and 73 have been mixed up. Obnoxious errors such as 'sotimes' (page 1), 'exceptions' (16), 'a author' (20), and 'its' instead of 'it's' (52) could easily have been caught.

Uniquely recommendable

One reason why *DITA 101* is one of the best books on the subject is that there simply aren't many others! *Practical DITA* by Julio Vazquez is a serious contender – and the perfect next step. Rockley and her fellow writers set out the strategy for you and make you familiar with the lie of the land at large. Vazquez helps you with the tactics down in the trenches, when you get up to your elbows in a DITA implementation. He first sets the stage with task orientation and topic-based authoring and topic reuse and then guides you from the bottom up with several examples.

In the end, the claim of *DITA 101* on the back cover is valid: 'If you're in the process of implementing DITA, or expect to do so in the future, or just want to learn more about it without having to wade through technical specifications, this is the book for you.'

I recommend *DITA 101* to anyone who is involved in a project to implement DITA, writing or translating documentation in a DITA environment or managing technical communicators. **C**

You can find more information at www.rockley.com/DITA101/index.html including the table of contents and a PDF preview of the introduction and Chapter 3, 'Reuse: Today's best practice.'



Kai Weber reports on a typical day where writing takes a backseat to collaboration and communications

It's an ordinary spring day, and I wake up in Frankfurt, Germany. When I'm lucky, the sun shines into my apartment to greet me. My morning routine is woefully brief, and I take a train to Bad Homburg shortly before 8am. The commute is pleasant since I live in the city, but work in the suburbs.

I'm a Senior Technical Writer for SimCorp, a Danish company that develops and markets the investment management system SimCorp Dimension to banks, insurance and fund companies.

I arrive at the office around 8:30am. I'm the only full-time writer in the German office, though a tester across the hall writes part-time. Ten writers are based in Copenhagen and Kiev.

I start up the PC and get a cup of tea. I receive an email from a colleague who has found an error in the online help. I check with the developer; my colleague is right. I check it against the Release Notes. This is embarrassing, they contain the same mistake, nobody had caught it. The Release Notes have gone out already, but at least I can correct the online help for the next release.

Next, I check the Solutions database, which is what we call our FAQ collection. We technical writers are responsible for editing the entries and aligning them with our documentation. A consultant has created an entry with a workaround. That's odd: the

Solutions entry does the trick, but we should really just fix that exception! I consult with the developer who will fix the bug. We will publish the Solutions entry for the time being and delete it in the next release.

It's 9:30am, and my testing colleague asks for my opinion. She is about to finish her first manual and has created a list of index terms and wants my opinion on them. Also, the Word template for manuals has become corrupted somehow, so we need to fix the page headers, which are supposed to show chapter numbers, chapter headings and page numbers. Her index looks good, so we agree on it very quickly. Then we wrestle with Word for a while.

At 10am, it's time for a video conference with the Copenhagen headquarters. I'm one of four people developing the future documentation strategy. We've already sketched out the processes on how we want to work in the future to implement structured and topic-based authoring.

Today, we discuss a design for an information model that I've drafted. I've basically taken a subset of DITA 1.1 and mapped its topic types and elements to our documentation contents. My colleagues have been reviewing it before the meeting, and they point out some parts that are inconsistent or confusing. Also, our model is still missing a couple of metatags, which means that my task until our next meeting will be to clarify some sections and to add the metatags. The video conference ends at noon, so I can catch my German colleagues as they head down to the cafeteria for lunch.

After lunch, I find an email by a colleague writer. I've agreed to review his manual, and here it is. The review is tricky since I don't know a lot about the module he describes. SimCorp Dimension is a fairly large and complex wall-to-wall system, so not every writer knows all modules in detail.

We had agreed that he'll need another reviewer for the actual contents, but I can still help him with the chunking of topics and the manual structure. I propose to change the nesting of topics in a couple of places. Also, the topic headings aren't fully consistent yet. I hope he will find my suggestions helpful.

Reviewing the manual reminds me that I still need to find reviewers for

my own manual that is just about finished. I can always count on the product manager (if he can find the time), but I like to have one of the implementation consultants review it as well. They know our customers and their workflows best from implementing the product on site, so their reality checks are invaluable.

At the same time, it's often tricky to find someone who can spend one or two days away from a project. It helps that they find the manuals generally useful. I approach the team leader and tell him when the manual will be ready and that it will take 6 to 8 hours to review. He has a couple of colleagues in mind, but needs to check their schedules. He will get back to me.

It's 3pm, and I start to actually write documentation. I continue to write Release Notes for the upcoming release. That means I go through all the development efforts in the tracking system and briefly document the enhancements and their benefits.

In between, I come across one enhancement where it is not clear yet whether it will actually be included in the upcoming release. I contact product management and, indeed, it's not been decided yet. That makes me nervous: in the previous release, some of these decisions came awfully late and required some last-minute editing.

Though I started writing rather late, I make some good progress. Around 5pm, I call it a day and take the train home. When I'm lucky, I can see Frankfurt's skyscrapers shimmer silver in the setting spring sun. **C**

Kai Weber is Senior Technical Writer for SimCorp GmbH in Germany. He has been writing, editing and translating software documentation since 1988. He has designed and implemented single-sourced documentation for corporate applications such as investment management systems, financial terminals, APIs and system architectures. He's a coach, trainer and mentor to technical authors.
B: <http://kaiweber.wordpress.com/>