WORDS

The beginning

Three years ago, I attended the TCANZ conference in Wellington, New Zealand. The waves were abuzz with Web 2 and how it might affect the technical writing profession. Some industry seniors who presented at the conference suggested that certain Web 2 products— especially wikis— might radically redefine the role of the technical writer. We might, they suggested, see our role move more towards content editors, away from our traditional role as creators of instructional materials. The bottom-up contributions of end-users (that is, customers) were considered a threat to the top-down approach of those especially skilled in technical communication. And some companies did indeed begin to delegate the task of writing end-user documentation to its customers.

I don't recall many technical writers publicly rebutting this new paradigm of relinquishing product documentation to the users of products. It seemed to me that we were so bedazzled, so headlight-frozen, by the new technology of Web 2 that we forgot the value that technical writers add to the increasingly important social responsibility of imparting practical knowledge. We seemed to have forgotten that those with language skills, an appreciation of user-centric writing and an awareness of the cognitive limitations on learning might be better able to impart practical knowledge.

And the professional bleeding continues. We now learn, paradoxically from a presentation that Adobe gave in London late last year to promote its new offering to *technical writers*, that the role of the technical writer is moving to that of *information curator* (see page 5). What? Curators don't create; they merely maintain. So we are, apparently, relinquishing editing as well as writing, becoming more like librarians and archivists. It's time to raise your voice against this rot.

Let me state yet again: our profession more than ever needs to emphasise, and re-emphasise, that what we have to offer is something much more substantial than what the average wiki-scribbler or I-obsessed blogger has to offer. Writing literate, audience-centric and efficient instructional materials requires specialist skills. Our skills. A bookkeeper is not necessarily best suited to writing bookkeeping instructions. To think otherwise is to abandon our profession altogether. It is professional suicide. Perhaps the Obama reforms (see page 11) will keep the profession alive. Here's hoping.

Geoffrey Marnell
Editor [geoffrey@abelard.com.au]

Contents

The single-source mindset

Lessons from a favourite childhood game

Gary Calwell

As a child, did you ever play a game of Chinese Whispers? If not, let me set the scene.

Assemble a small group of children and ask them to form a line. Ask the first child to think up a thirty-second story and then whisper it to the second child. The second child whispers the story to the third child and so on until finally the last child writes down the story as they heard it.

What are the chances that the final story will be in any way similar to the original? That's right: almost zero.

If we explain that the objective is to keep the story as accurate as possible along the chain, you would expect that after several attempts the quality of their communication would improve. Now let's make the game more interesting. Offer to give each child a sweet for every minute that they play the game. The children now have two competing motives: a team goal to finish the game quickly by keeping the story accurate and individual goals to prolong the game as long as they can get more sweets.

What does this game have to do with technical writing and the single source mindset? More than you might think. When these children become adults, they might find themselves working on a large project. If so, chances are they will get to play a

grown-ups' version of this game. The whispers might be replaced by spreadsheets, status meetings and reports—and the sweets might be replaced by daily rates and contract fees—but the competition between team goals and individual goals often remains.

In Chinese Whispers there are many versions of The Truth, each represented by each child's version of the story. This is also true of many large projects, where different teams work in silos to create their own "story" and try and keep all the stories aligned by "whispering" to the other teams. Let's look at a real-life example.

A company is attempting to develop and deploy a new customer management system. We'll count how many versions of The Truth there are in this multi-million dollar project. The business analysis team writes design documents that are a mixture of technical information and end-user procedures (1). Since there are no defined standards, the output predictably is very inconsistent. The technical writing team uses the design documents to create a set of user manuals (2). Since some of the information in these manuals is not suitable to be used in training, the training team creates a set of training materials (3). The testing team creates a set of test scripts (4). Finally, the operations team creates a set of work instructions (5).

> Single-sourcing is a mindset, not a software feature

We now have five sets of information and five versions of The Truth. Each set of information is designed for a different purpose but there is considerable duplication of content. If you need any proof of this, just look at the most common authoring process: copy from one document set, paste into another and edit as required. In theory, a process like this can work. Information can flow from one silo to the next through well-defined entry and exit gates. In practice however, this process usually fails, for it is based on two myths:

- Myth 1 is that information flow is one-way. Authoring systems are designed so that information flows from one document set to another. In practice, there are many feedback loops as teams make design changes and fix errors. Having feedback and rework loops in a single set of documentation is hard enough to manage, but try doing this with five sets.

- Myth 2 is that documentation will always be kept up-to-date. However, it is usually difficult to get others interested and motivated enough to keep one set of documentation up-to-date. If a project team is struggling to achieve its milestones, the task of updating documentation is often pushed down the priority list, especially if there are multiple sets of documentation to keep track of and maintain.

The concept of single-source documentation is simple enough. The content in one set of documentation can be filtered, sorted and formatted in different ways so that it can be used for multiple purposes: for example, a user guide, technical manual and training guide could be produced from a single document. Many people think of single-source as a software feature rather than as a process of eliminating content duplication regardless of the technology used. If you visit the web sites of help authoring tools, for example, you will see that a "feature" of their software is its ability to produce multiple output documents from a single set of source documents.

However, you can achieve a single-source solution using the least sophisticated of technologies. Imagine a group of product managers who need to give presentations about a new product to different audiences. Using a blank sheet of paper and a pen, they could discuss, clarify and write down the key points of the presentation. Using different coloured highlighters they could then mark which points were most relevant to each audience. There you have it. A single-source solution for under $10!

Single-sourcing is a mindset, not a software feature. It's about allowing a group of people to collaboratively work on a single document set that can later be used to meet all the project's downstream information needs.

So how could we apply these principles to the earlier real-life example? The first step is to clarify the purpose and audience of each document, eliminate duplication and use hyperlinks to cross-reference rather than copy and paste. We could also encourage greater collaboration between the teams. For example, there is often similarity between the scenarios used for testing and training exercises. Why not use the same scenarios for both purposes?

If you think in these ways you might come up against the limitations of your authoring tool (Microsoft Word 2007 in our real-life example). If so, you may be able to use the automation features of your authoring tool to make it easier to do repetitive tasks. If you want people to change their behaviour, it's much easier if the new way is easier for them than the old way. In our real-life case we used Visual Basic for Applications (VBA) macros to add new functions, such as:

- A new hyperlink command that allows users to link to any topic in a user manual without the time-consuming need to manually insert bookmarks in the destination document.

- Locking all documents—other than tracked changes—so that users don't corrupt the document's styles and layouts. When an authorised user (from the technical writing team) opens the document a macro automatically unlocks the document and locks it again when the document is saved. This provides a reasonable level of security but, more importantly, allows everyone to collaboratively work on the same document set.

- Adding text tags to the Microsoft Word documents so that they can produce both facilitator guides (containing facilitator notes) and participant guides (excluding participant notes).

Customisations like these take some time (and therefore money) to build and maintain and so it is important to do some form of cost–benefit analysis. For small projects with a small team it is probably more cost effective to continue to do things manually. But for a large and complex project, and a large team, automation can result in substantial benefits including reduced development time, better quality and greater consistency—not to mention dramatically reducing everyone's stress levels.

So, the next time you walk out onto the project floor, think about to what extent the different teams are really embracing single-source. Listen carefully for the whispers and look for duplication in your documentation sets. Once you become aware of these problems, it's then possible to act on them. In doing so, we can add real value to the project and help others see that technical communication really can be an asset to the project rather than a liability.

Gary Calwell

Gary Calwell is the director of Claricom Consulting, a Melbourne-based consultancy specialising in process improvement, documentation, training and Microsoft Office customisation. He can be contacted at gary.calwell@claricom.com.au.

From product knowledge to knowledge product: The technical writer connection

Raymond Calbay

By day's end, you might think that the few pages of help topics you've written don't look like much of an accomplishment. But consider this: you've put in effort learning how the product works, deciphering what you could from cryptic functional specifications. You've also collaborated with subject-matter experts to explain in simple language an otherwise complex technological beast. Definitely, your five pages entitle you to a pat on the back.

What's more, it isn't long before your instructions and concepts build up to a complete book (or a library even). Whether read by colleagues or by mainstream consumers, the value of your work lies in how you've distilled and shared information that could have been potentially lost without consideration. This is the stuff that MBA types talk about when they drop the term "knowledge management".

Knowledge management (KM) treats the skills, competencies and insights of people within an organisation as the most important business asset. Since knowledge could walk out the door at any minute, a strategy should be in place to encourage people to pass on knowledge critical to the business's success. Put another way, knowledge management is "the process through which an enterprise uses its collective intelligence to accomplish its strategic objectives".[1]

Indeed, is there anything more important for businesses than to make customers happy, and therefore gain loyalty? The first criterion has always been whether customers could easily and effectively use a product. And it's in the documentation you create where they'll find information about the correct and safe use of their newly owned gadget or

1. R Barquin, "What is knowledge management?", *Knowledge and innovation*, vol. 1, iss., 2, 2001, p. 128.

software. Sometimes, the first thing they'll see when they unpack the product box is your quick start manual.

Let's take a closer look at how the technical writing process parallels the KM cycle.

Knowledge gap identification

Suppose that a work request for a new user manual lands in your inbox. To help you plan an outline, you start looking for any information you can get your hands on. Your research takes you through dozens of folders on the shared network. But at least you have access to a few critical sources: development designs, business cases and a directory of the project team to contact for questions. You review these materials to assess who might be the end-users of the product. Having a clear picture of the end-user in mind narrows down your approach in writing potential topics. You wait for a product prototype for you to test and tinker with. At this stage, you've identified crucial gaps in your knowledge production process: the information sources, the key people to coordinate with and the technology to learn.

Knowledge capture and creation

After researching and familiarising yourself with the product, you've gained the confidence to explain it concepts and procedures. Except for a few clarifications you need from the development team, you're ready to roll out your draft. During an interview, an engineer demonstrates how to use a particular feature that he doesn't know how to put in words. You jot it down as a hint that you should probably include in your topic. This is an example of what Dr Melissie Rumizen describes as tacit knowledge—"what we do not know that we know" —becoming explicit knowledge, and therefore recordable.[2] Information that cannot be easily written down often surfaces when you talk to subject-matter experts or join planning meetings. As your writing and revising continue, you take part in the "codification of knowledge".

Knowledge validation

Since you've used information from numerous documents and resources, you need to submit your draft to a technical review. This process ensures that the overviews and instructions you've provided are correct and that you haven't missed any crucial step. Your interactions with engineers, quality analysts and planners also guarantee that their expertise gets into your work. You can even get to facilitate if their

ideas conflict with one another, as Michael Hughes recommends.[3] He says that technical writers "help design teams arrive at consensus about what the product is or does". Be wary though that since reviews are often subjective, you might need to ignore or challenge some of your reviewers' comments. It may be that your product manager wants you to change several topic titles. But if you know that you've got it right, especially if it concerns writing style and tone, don't hesitate to assert your expertise. Armed with your style guide, teaching people a thing or two about your domain knowledge is never a bad thing.

Knowledge distribution and application

Your content is now ready for production. You upload your revisions and check your source files into the content management system. Since you author in XML, you output in several formats: a print-bound PDF book, an HTML help system and a set of chunked web articles. The development team integrates your deliverables in the final package. As the product hits the market, the knowledge you've captured also gets delivered. Don't fret that your manual is likely to be tossed out. In various contexts of product use—from setting up to troubleshooting—customers will need some instructions and would hold your company to your word. Now isn't that overwhelming? No matter, your manual will develop a life of its own. Elements of the template design, cross-references and even the index terms you've used will serve as entry points for more information. When customers figure out how to work the product from reading the manual you've diligently prepared, that's your contribution to knowledge management.

Have you heard the saying "knowledge begets knowledge"? Pretty soon, you'll receive emails from readers: from customers or from staff within your organisation. Their comments won't necessarily be about errors in your work. They might also be suggestions for improvements, thus providing you with an opportunity to assess gaps in your current user manual—your very own knowledge product. Check your inbox because the knowledge cycle keeps on going on.

Raymond Calbay

Raymond Calbay is a content specialist based in Taiwan. He holds a masters degree in communications from the University of Santo Tomas in the Philippines. He presented research on the intersections between KM and technical writing at the 2010 Asian Media and Communication conference in Singapore. You can contact him at raymond@halogenknows.com.

2. MC Rumizen, *The Complete Idiot's Guide to Knowledge Management*, Alpha Books, Indianapolis, 2001, p. 8.

3. M Hughes, "Moving from information transfer to knowledge creation: a new value proposition for technical communicators", *Technical communication*, Aug. 2002, p. 280.

Adobe's *Trends in Technical Communications*

Dan Smith

Adobe ran an event titled *Trends in Technical Communications* at their London office on 18 November 2010. The event was designed to promote the release of the new *Adobe Technical Communications Suite*—consisting of Framemaker 10, Adobe AIR 2 and a new version of Robohelp—by pointing out how the *Suite* meets the needs behind the current trends in technical communication.

The first of four sessions covered the five main trends in technical communications. These are:

- Movement from unstructured documentation to structured documentation, specifically documentation structured according to DITA. This is driven by the desire to reduce localisation costs and, to a lesser extent, by the desire to take advantage of the savings implicit in content re-use and in simplified documentation development.

- An increasing amount of content delivered as multimedia at the expense of traditional written content, and for this multimedia content to be accessible from any platform. The speaker quoted the prediction that by 2015, access from hand-held devices (iPads, iPhones and Android phones) will surpass access from conventional PCs, laptops and netbooks.

- Optimising for large quantities of information, primarily by improving ease-of-access and searchability. Online searching is becoming the primary method of locating information. Hence, content needs to be optimised for Google Search. It also needs to be optimised for personalisation (which can be done using metadata). For example, users should be able to restrict the content offered to them based on their role, their country or some other attribute.

- Increasing use of rich media, such as PDFs with embedded videos and embedded interactive tutorials.

- Collaborative content creation, involving collecting user feedback and fostering social networking to promote inter-user collaboration. As a result, the technical communicator's role moves towards an information curator role. *Information curator*. Remember that term, and add it to your collection of potential buzz-words.

- Increasing demand for single sourcing, with a convergence of development tools. This wasn't spelled out, but I took it as a good omen for content editors such as FrameMaker and XMetaL at the expense of products like RoboHelp.

Ankur Jain, the next speaker at the presentation, described how FrameMaker 10 is designed with all these trends in mind. With the advances in FrameMaker's features, Adobe is obviously serious about pursuing the CMS and DITA market. I was quite impressed with FrameMaker 10, although I have to confess to being a FrameMaker fan from way back. One of the things that impressed me was that FrameMaker uses DITA-compliant metadata to implement conditional text.

You can now publish to a variety of output types directly from FrameMaker, including AIR—Adobe's new bells-and-whistles output type. It looks likely that if you wanted to use the Adobe publishing suite, you could use FrameMaker for all your publishing.

The next speaker—Peter Grange—spoke mainly about RoboHelp. It was interesting to hear about .chm and WebHelp. I didn't really know the history, but the .chm format was effectively killed off by a Microsoft security patch in 2005. The patch prevented a .chm file from being run over the web. Since then, the .chm format hasn't had any updates. The WebHelp format, on the other hand, has been and continues to be updated. Look-and-feel options and functionality have been improving and are expected to continue to improve.

Peter also spoke about Adobe AIR, which is Adobe's online help delivery product. The current

version allows for adding comments to a shared network location. The next release, however, will enable user comments to be saved to a web location. This means that external customers and not just internal users will be able to add and share comments. The functionality at present only allows for unformatted text comments. I suspect that Adobe AIR is the product that Microsoft is trying to catch up to with Help Viewer. Like Help Viewer, Adobe AIR lets you download content or access it online. It also notifies you of updates.

The last speaker was Julian Murfitt from Mekon. Julian provided a cut-down version of his presentation to the last STC conference. It had some interesting points, such as:

- The *Goldilocks syndrome* is an expression coined to describe the apparently increasing behaviour of corporations of trying a few types of system before settling on the one that best suits them.

- The importance of metrics so that documentation teams can benchmark and quantify improvements. Julian said that for one organisation his consultancy worked with, they quantified the documentation development cost at £65 a page. (Apparently management understands pages, so that's a convenient measure.) Now it is possible in that organisation to say things like "If we do x it will reduce costs to £60 a page" and "If we do y it will increase costs to £70 a page, but it will deliver these benefits".

- The importance of putting in the effort to find out what customers really want from the documentation, and it isn't always what you might think. But of course we already knew that.

- When Mekon works with organisations, one of the things they aim for is getting all stakeholders involved and on-board. It's not uncommon that some stakeholders find it hard to come to grips with XML and DITA, and to understand the benefits and possibilities of an XML–DITA environment. They sometimes have to spend considerable time in conveying the mechanics and what it makes possible. This highlights the extent of the generational leap from pre-web to post-web. It's easy for those of us closely involved to under-estimate just what a paradigm shift authoring in an XML–DITA environment is.

Julian asked for a show of hands of those who were familiar with DITA. Interestingly, only half of the attendees raised their hands. Considering that probably a lot of those have only Googled the term, I thought the level of familiarity was surprisingly low—especially given that the conference content was so DITA-oriented.

So to sum up, it looks as though Adobe is embracing the DITA standard—which makes it pretty much a clean sweep across the industry—and it looks likely that the DITA format will at least complement, if not drive, the way documentation evolves.

Dan Smith

Dan Smith is an expatriate Australian working in the UK. He has many years experience as a technical writer, working with large technology companies in Australia, Italy, France and the UK.

The Language of Marketing

Lana Brindley

I was absently staring at a new tube of toothpaste this morning as I washed my hair. You have to look at something, right? This one declared "healthy, whiter teeth for longer". An image of extremely long (but healthy and white) teeth filled my mind, and was immediately pushed out by the technical writer in me asking "whiter and longer than what, exactly?"

Most marketing slogans give technical writers the screaming heebie-jeebies. Not only do they make spurious and vague claims like "more fibre", "less fat" and "20% bigger" with alarming regularity, but the adjectives! I have no doubt that it is actually possible to sell things with sentences that contain only one adjective. And if they do need more than one, I'm sure a comma wouldn't kill them.

I could rant on the folly of adverbs, too, but that is a different article.

Why are marketers such terrible writers?

Because customers expect spin, and spin is easy to write. All you need is a handful of adjectives and a call to action: "The new fruity refreshing Globswoddle Fizz is now available. Experience the heady taste of summer today!"

While yelling at the toothpaste tube in the morning might make us all feel better, it is not likely to turn us into marketers just to help an obviously flailing industry. I finished my marketing degree about three weeks before I decided that the marketing industry was the last place in the world I wanted to work. Eventually, I became a technical writer instead, and discovered that I had inadvertently ended up working in marketing after all. Every word we set to paper is marketing in one way or another. If it is going to be read by a customer, then it needs to sell the product. But the last thing we want to write is spin.

Why are writers such terrible marketers?

Because customers want anything but spin, and while spin is easy to write, spin-less marketing is not so easy.

Spin is wanted and welcomed in places where it is expected, like product packaging and on the airwaves. When our customers read technical manuals or help text, they are looking for a solution to a problem. If they were suddenly faced with the empty promises of spin, they would lose faith in the documentation, and possibly the product.

However, brutal honesty is not required, either. Product documentation should not tell customers that the product cannot fulfil their expectations. Every question needs to be anticipated and answered. The documentation must give the customer hope that their problem can be resolved, their task completed and their sanity retained in the process.

Effective documentation never tells the customer that a product is terrible (even if it is), and it never tells a customer that they are stupid (even if they are). It never makes over-inflated claims of software brilliance, and it never assumes greater-than-average user intelligence.

Somewhere nestled in there is product documentation that shows the product in a positive light without the hard sell. Sound easy? Like most technical writing, it sounds easy until you actually

try to do it. Some tips for getting started with spin-less writing:

Kick adverbs, take names

Adverbs are a big red flag for spin. Be ruthless and cut them all out. If your sentence requires a modifier, consider what you are really trying to say. If it forms part of an instruction or description ("The widget can be fully removed by …"), reword it to remove the adverb ("Remove the widget by …").

Never call anything "simple"

If you tell your users that something is "simple", "quick" or "easy", and the customer struggles with it (for whatever reason), you are essentially telling them that they are failures. Try not to insult your users.

Mind your adjectives

Adjectives are fine in their place. Use them only where necessary, though, and try not to use more than one at a time. ("Locate the red button" is fine, but avoid "Locate the large, shiny, red button" that is next to the "tiny, silver, shiny lever").

Know your stuff

If you can't describe your topic in a single short sentence, you don't understand it well enough, and it becomes too easy to succumb to spin statements. You need to be able to give succinct and accurate descriptions for each and every component, as well as the product as a whole. If you are not able to do that, continue to research your product until you can.

Understand the enemy

As modern humans, we are largely desensitised to advertising, simply because we are so totally immersed in it. Start noticing it. Analyse what language is used and the sentence structure employed. Work out how you would re-write it to send the same message, but without the spin.

Edit with a knife

Never say more than you need to.

Lana Brindley

Lana Brindley has been playing with technology since that summer in the 80s when she spent the whole time trying not to be eaten by a grue. She has been writing since she could hold a pencil, and is currently writing technical documentation for Red Hat. Lana holds business degrees in marketing and information systems, and with any luck will have a technical communicators degree by the end of the year. She works from her home in Canberra, Australia, and occasionally leaves the house in order to berate university students and conference goers about passive sentence construction.

Ethical dilemmas in technical writing

Real-life challenges at the coal-face

Suppose that a company's Customer Support Centre (that is, the department that answers queries from customers) is expected not just to help customers with their queries, but to generate revenue for the company by selling training courses to the customers who contact them. The manager of the Customer Support Centre tells you, the lone technical writer for the company, that good technical documentation (that is, documentation that clearly explains how to install and use the product) denies the company an opportunity to create revenue from training courses, since a customer who reads that documentation will have no need to call the Customer Support Centre for help.

What would you do? Write the best documentation you can? Or introduce the occasional vagueness, error or critical omission so that customers will need to ring the Customer Support Centre for help and be coaxed into purchasing a training course?

What if the manager of the Customer Support Centre is also an official reviewer of your work and asks you directly to change a procedure you have written so as to make it likely that a customer who wishes to follow that procedure will need to call for help?

What *would* you do?

Journals

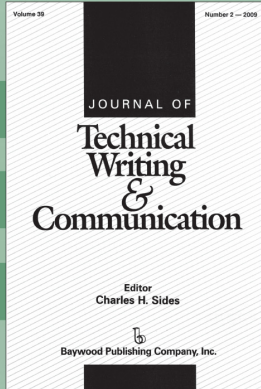## Journal of Technical Writing and Communication

Contents of the current issue

Click the heading below to read the editorial, abstracts and reviews, or to purchase an article or subscribe.

Volume 41, Issue 1, 2011

- Warp and Weft: Weaving the Discussion Threads of an Online Community, Lora Arduser

- The Communication Habits of Engineers: A Study of How Compositional Style and Time Affect the Production of Oral and Written Communication of Engineers, Derek G. Steiner

- Evaluating Applications for an Informal Approach to Information Design: Readers Respond to Three Articles about Nursing, Russell Willerton and Mary Hereford, RN CEN

- Linking Contextual Factors with Rhetorical Pattern Shift: Direct and Indirect Strategies Recommended in English Business Communication Textbooks in China, Junhua Wang and Pinfan Zhu

Book review

Reviewed by Suzanne Pearce

In this collection, academics and professional writers delve into the world of content management systems. The knowledge economy's greatest asset is *information* and is primary problem is information *management*: finding it, validating it, re-purposing it, keeping it current and keeping it safe. In the last few years, content management software has become as common as word processing software. But unlike word processors—which are designed for solo authoring and local storage—content management systems are designed to accommodate large-scale information production, with many authors. Each author provides many different pieces of information kept in a web-accessible database, any piece of which might find its way into electronic documents that the author doesn't even know exist. These software systems are complex and their impact on the field of writing will be immense.

Content management systems are rarely chosen by the people who will be providing the content. It is not just the content authors who should be concerned at the implications of such systems. More than ever, senior management and IT professionals need to understand what it is that technical communicators do. They need to understand that technical communicators offer more than just technically correct paragraphs describing some product or other. For content to be usable as well as re-usable, the implementation and use of a content management system needs to be an exercise in understanding and cooperation.

According to the publishers, "this book begins a conversation about the effects of electronic content management systems on the profession of technical communications and advocates the need for more community involvement in the design and implementation of CMS". The book was largely written by academics in the education and training of technical communicators, as well as practitioners in the fields of technical communication information design and business communications. The advice given will be of most interest to other academics and trainers seeking to incorporate content management systems into their teaching methodologies.

However, on reading this book, the students who are being taught by these writers may find themselves wary of their choice of career. They are advised of some of the difficulties of implementing and using large academic and corporate content management systems when applied to either courseware or the help system environment. They are told of the potential to return to the drudgery of simply writing correct blocks of text with no opportunity for input into design and style. They are also warned of the legal ramifications of outsourcing the writing of that same text, particularly when the outsourcing goes offshore and is possibly then subcontracted to a cheaper organisation that may be in yet another country. It tells students that they will no longer be assured of employment based upon expertise in authoring and web design, and that they should be prepared to continually upskill themselves. For example, they will need to consider learning the XML that readers are advised should be taught as part of technical communication curricula.

Though the audience for this book may be appear at first small, the book is still worth a read by all who have an interest in the future of technical communication in a world where content management systems are widely deployed.

Suzanne Pearce

Suzanne Pearce is an Adelaide-based technical writer.
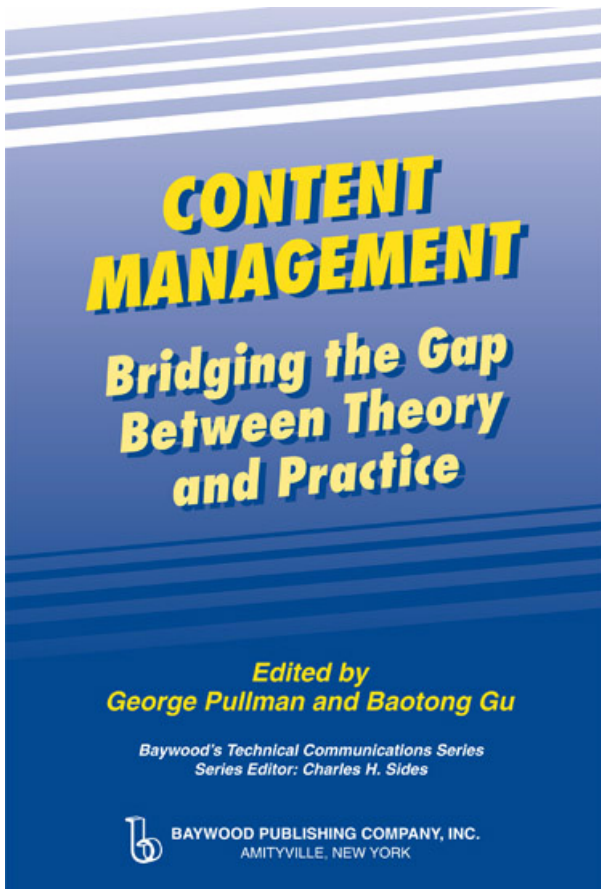
Tips and Tricks

Touching up a PDF file

Ideally, errors found in a PDF file should be corrected in the source document and the PDF regenerated. But sometimes that is not possible. The source files might be missing, the owner might be out of reach or the correction needed might be beyond the skills of those who are available.

In the early days of PDF files, there was not much that could be done by way of editing a PDF file. And that was considered, then, one of the virtues of the PDF format. Things are much different nowadays, and there is much you can do to a PDF—providing,of course, it is not password-protected against changing. And providing, too, that you have the full version of Adobe Acrobat, not just the Adobe Reader.

Changing text properties

The **Advanced Editing** sub-menu (accessible from the **Tools** menu) provides many useful tools for touching up text and graphics. Many of us, no doubt, have used **TouchUp Text Tool** to correct the odd infelicity: a spelling mistake, or a missed word or out-of-date information. But you can do a lot more than just add or change a character or two. You can, for instance, change the font, the font size, the colour (of the fill and the stroke), the stroke width, character spacing, word spacing, stretch and baseline offset. To do so:

1. Select **Tools** > **Advanced Editing** > **TouchUp Text Tool**.

2. Select the text you want to change.

3. Right-click on it and select **Properties** from the context menu.

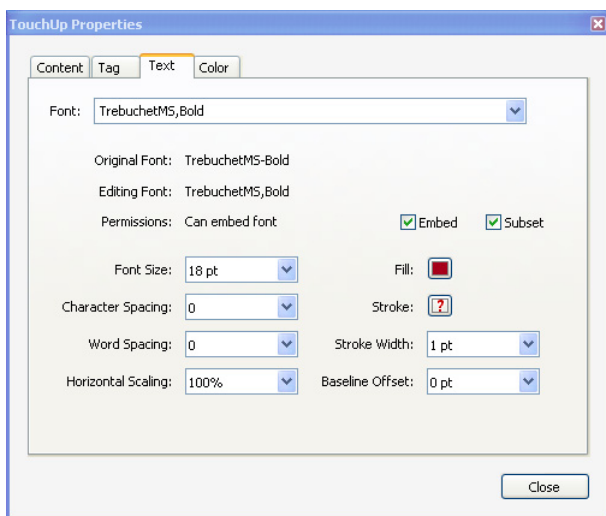   The **TouchUp Properties** window appears (see figure 1).

Figure 1: Setting new text properties

4. Make your changes and click **Close**.

Adding text blocks

You can also add a new text block with **TouchUp Text Tool**. You can do this with the **Typewriter** tool too (also accessible from the **Tools** menu), but the range of text properties you can control is greater if your use **TouchUp Text Tool**.

To add a new text block:

1. Select **Tools** > **Advanced Editing** > **TouchUp Text Tool**.

2. Hold down the CTRL key and click where you want the new text block to start.

   The **New Text Font** window appears.
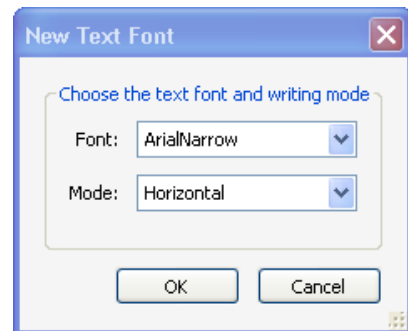
3. Choose a font and orientation for the new text block and click **OK**.

4. Start typing your new text.

   You can press ENTER to start a new paragraph.

5. If you want to format any text, select the text—**TouchUp Text Tool** must still be active—right-click on it and select **Properties** from the context menu.

   The **TouchUp Properties** window appears (see figure 1).

6. Choose the text properties you want and click **Close**.

Jumping to the previous cursor position in a Microsoft Word document

It is a common occurrence. We are typing away in Microsoft Word, leave what we're doing to add to or edit some other section of the document, and then return to what we were doing—if only we could remember exactly where we were before. We invariably find where we were, but wouldn't it be nice if we could jump straight back there without much effort.

Well, a not-too-obvious feature of Microsoft Word enables you to do just that. Press SHIFT + F5 to force the cursor, and focus, straight back to where it was before. In fact, Microsoft Word remembers your last three cursor locations. Repeatedly pressing SHIFT + F5 cycles between these locations.

Miscellany

Legal Plain English

It took 20 months to pass through both houses of parliament and remarkably had 33 against it when it was voted on in the US House of Representatives. But finally, on 13 October 2010, President Barak Obama signed into law *Plain Writing Act of 2010*. The law is to take effect from 11 October 2011.

The purpose of the Act is to improve the effectiveness and accountability of government bodies by insisting on the use of plain language in communications directed at the general public. It defines plain language as writing that the intended audience can readily understand by being clear, concise and well organised.

You can get an idea of the language being decreed by visiting http://www.plainlanguage.gov/. Most of the recommendations you will find at that site are eminently sensible, and technical writers might benefit from reading the guidelines to refresh their understanding of good, audience-centric writing. Be wary, though, of taking the advice implied by section headings without reading the entire section. For example, *Use active voice* is a section heading and would imply that you should always use active voice. However, the very last paragraph of the section provides reasons for preferring the passive over the active. It's not all what it seems.

Shooting passives with blanks

And while on the topic of passive and active voice, here is some research that proves, well, nothing at all, really. In a paper entitled "Use of the passive voice in medical article journals", Professor Robert Amdur and colleagues compared the incidence of passive voice in three medical journals (20–26%) with the incidence of passive voice in articles on the front page of *The Wall Street Journal* (just 3%).[4] On the basis of such variation, the authors argue that "medical journal editors should make a passive voice frequency of ≤ 10% a publication requirement for all types of articles".

First of all, why 10% and not 15, or 5? Anyway, how can one compare journalistic style, with its emphasis on exposition, with scientific writing style, with its emphases straddling methodology and conclusion? It's chalk and cheese. Miscellany awaits a parallel article in *Technical Communication* that recommends that, since the incidence of rhyming verse in modern poetry is about 30%, technical writers should limit their use of rhyme in end-user documentation to 30% (or whatever).

New Words in American English

The *New Oxford American Dictionary* (affectionately referred to by the acronym *NOAD*) was republished in August 2010 (being its third edition). In the five years since the release of the second edition, 2 000 new words have been deemed worthy of inclusion (deemed worthy on the basis of having reached a critical mass of common usage). Common usage or not, I suspect that many of these new words will not find their way into technical writing, words such as:

- *lipstick lesbian* n., a lesbian who favours a glamorous, traditionally feminine style

- *tramp stamp* n., a tattoo on a woman's lower back

- *wardrobe malfunction* n., an instance of a person accidentally exposing an intimate part of their body as a result of an article of clothing slipping out of position

- *homeshoring* n., the practice of transferring employment that was previously carried out in a company's office or factory to employees' homes

- *bromance* n., a close but non-sexual relationship between two men

- *truthiness* n., the quality of seeming or being felt to be true, even if not necessarily true

- *heart* v., to like very much; to love, as in "I totally heart Boston"

Mind-googling translations

When the weather is foul and the television dull, you can improve your spirits with a game or two of Google Translate. It's easy. Start with a sentence in one language, pass it through the Google Translate mangler a few times and then bring it back into the original language. For instance, enter "Technical writers are a valuable asset to any organisation", translate it into Arabic and then translate the Arabic back into English. The result? "Art book and a valuable asset to any organization". Sort of.

Another example: start with "Technical writers must always write in ways that are maximally familiar to their intended audience", translate it into Japanese, translate the Japanese into Korean and then bring the Korean back into English. The result? "Always known as a technical subject in a way that must be created". You see? Hours of fun for the entire family.

And one more for good measure: translate "hanging hyphen" into Polish and then translate the Polish (wiszace lacznik) into English. Ah, I should have guessed that a hanging hyphen is a … *wall switch*!

---

4. *AMWA Journal*, vol. 25, no. 3, 2010, pp. 98–104. This is the journal of American Medical Writers Association.

Mindstretchers

Geoffrey Marnell

A testing time for all

After much pestering by her students for the results of a recent test (where the maximum score was 50), mathematics teacher Gwendolyn Prime decided to relent just a little by telling her impatient class that:

a. the product of the best mark and the lowest mark is 1012

b. six times the average of the best mark and the lowest mark equals the sum of all the marks together

c. half the sum of the second best and third best marks equals twice the difference between the best and third best marks, and

d. all marks—no two of which are the same—are whole numbers.

"Armed with this information", she told her charges, "you should be able to work out the marks I have awarded". Can you?

Last puzzle

The puzzle in the last issue of *Words* posed the following:

I hold my hand between a strong lamp and a wall so that a noticeable shadow appears on the wall. I then move my hand parallel to the wall for a little way. Does the shadow of my hand move across the wall more slowly than, faster than or at the same rate as my hand moves?

Solution

The shadow will move faster than my hand. The diagram below shows that the distance travelled by any point on the hand is always less than the distance travelled by the corresponding point of the shadow.
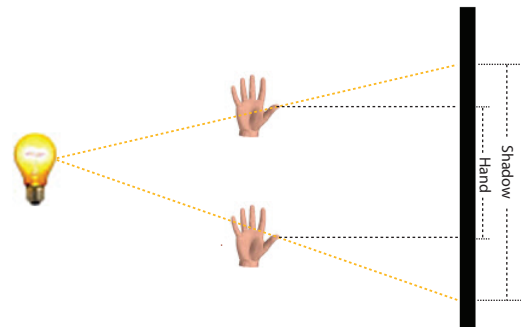
Now, the shadow only moves while my hand is moving. Since the speed of an object equals the distance travelled divided by the time taken, the shadow—*in going further in the same time*—must have travelled faster than my hand.

Contributions Wanted

If you would like to contribute an article or review to *Words*, send an outline of what you would like to contribute to the editor at words@abelard.com.au.

Words is published four times a year: in February, May, August and November. Contributions must be on a topic of some relevance to technical communication.

Rough drafts will not be considered. You should only submit a draft if it is in a state that you consider is worthy of being published.

The editor reserves the right to make editorial changes, but will not publish your contribution unless you have agreed to any changes that have been made.

Copyright is retained by contributors.

The *Words* team

- Artwork: Christine Weaver
- Copy-editor: Marcia Bascombe
- Editor: Geoffrey Marnell
- Contact: words@abelard.com.au