

WORDS



A Quarterly Bulletin for Technical Writers & Communicators

Volume 2 | Issue 4 | November 2010

The beginning

Myths about our profession—and about why we do certain things and not others—are many and varied. Ten years ago, Geoff Hart—a Fellow of the Society for Technical Communication—wrote about ten such myths in the hope that they would be banished to the scrap-heap of history. Alas, as Geoff pointed out in giving his agreement to publish an updated version of his original article, these myths are still prevalent today.

One such myth is that technical writers add no tangible, or quantifiable, value, that we are a mere cost centre. Despite all the efforts of our various professional societies, this is still the view of most employers of technical writers in Australia and North America. It is much the same in Europe where, as Dan Smith writes in his article on the changing world of technical writing, “documentation is [increasingly] seen purely as a cost area, necessary only to meet basic user information obligations”.

Our reputation is certainly in decline. And it's not all the result of the global financial crisis. More and more companies—large and small—are outsourcing their documentation, and to an unlikely source: their customers. Take Adobe and Nokia for example. Their products are accompanied by miserly documentation, the expectation being that customers who want to learn how to do what the product marketing tells them the product can do will find a user forum and pose their questions to it. Ever wondered how to combine Adobe RoboHelp projects into one AIR deliverable? Or wondered how to set up your Nokia phone to receive emails? Forget the product documentation. You will find no instructions there. The user forums are your best bet.

Industry's unflattering opinion of our profession is being expressed in even subtler ways. It is industry that, in the past, has mostly paid the tuition fees of employees who wanted to gain a qualification in technical writing. Alas, enrolments have been shrinking, so much so that Swinburne University—until now Australia's only university offering a qualification in technical writing—is no longer taking enrolments in its technical communication courses (see page 20).

Could we be partly to blame? Roll on accreditation.

Geoffrey Marnell

Editor [geoffrey@abelard.com.au]

Contents

Ten technical communication myths	1
Font choice revisited	9
The changing world of product documentation	11
Numbering in FrameMaker 9 books	13
Journals	15
Book reviews	16
Tips and tricks	19
Miscellany	20
Mindstretchers	21
Volume index	21

Ten technical communication myths

Geoff Hart

Myths often represent the very human attempt to explain something important but poorly understood, such as the turning of the seasons, or to provide cautionary tales to warn listeners against unsanctioned behaviour, as in the myths of Prometheus and Epimetheus. The fascination inspired by myths has kept many alive across the millennia, but despite the degree of abstraction or exaggeration that makes them so fascinating, there often lies a grain of truth or an insight into some fundamental aspect of the human condition at their heart. In our current enlightened age, we fancy that we've grown beyond the need for myths, yet urban legends abound (particularly on the internet), and many of the things we do in our daily work are strongly influenced by rules of thumb that are, in a very real sense, a form of myth.

Like any other profession, technical communication has accumulated its share of mythical rules of thumb, but the good news about our profession's myths is that they too contain grains of truth and insights into things that are truly important to us. The bad news is that we've also internalised some of these myths to the point that we no longer question them, and have begun to let them constrain our choices rather than to help us remember and see the truth. Some communicators

even overgeneralise the occasional rule to the point at which it loses its validity and become dangerously misleading.

So what myths do we live by? In no particular order, this paper presents my top-ten list of what I consider to be the central myths in modern technical communication. There are undoubtedly others. By writing this paper and acting as devil's advocate, intentionally presenting these myths in a bad light, I'm hoping that I can persuade you to question these and other rules of thumb that you use daily. When you pay closer attention to the rules you obey, consciously or otherwise, and question why, you can start to recognise the disabling aspects of a myth and begin taking steps to free yourself from those constraints.

Myth 1: Knowledge of specific tools is vitally important

Few managers want to hire a new technical communicator and wait weeks for the person to become productive with the company's writing tools, yet hiring on the basis of tool skills ignores the fact that the ability to format text is a very small part of our value as technical communicators. (It also ignores the fact that any new employee, even one who comes equipped with the desired tool skills, faces a learning curve at a company and may take weeks to learn the ins and outs of the new job.) Employers hire us primarily because we can understand their products and communicate that understanding to their customers. They hire us because we know how to take a product apart, literally or figuratively, and decide what components of the product we must document and how we should do it. They hire us because we possess the ability to pry information from the grasp of reluctant subject-matter experts, because we have that rare skill of empathising with our audience well enough to understand their needs, and because we have the persistence to eventually satisfy those needs.

None of this depends strongly on the ability to work in Word, Framemaker or RoboHelp. Back in the Dark Ages before computers, the ancients did a pretty good job of documenting complex things without these tools; in fact, those ancients could probably teach us a few things about good writing. Nowadays, few writers lack the ability to type and do basic formatting from the software's menus, and these (not formatting skills) are the crucial tools that support our work. In many situations, advanced formatting skills are actually a red herring, because templates already exist and layout or design work consists more of applying the templates than of actively designing something new. It's not that knowing how to format is unimportant to us; rather,

it's far less important than our ability to communicate.

But let's assume that tool skills really are as important as some managers claim. Given that most of us have learnt enough software skills to quickly develop basic to moderate competence with new software, the period of several weeks while we adapt to our new job is more likely to pose problems than our ability to learn new software. For example, in my career (now over 25 years), I've mastered four layout programs, half a dozen word processors, three operating systems, and more other programs and utilities than I care to count, all the while coping with an ever-accelerating rate of evolution in each of these software categories. What's impressive about this is not that I'm a software prodigy, but rather that I'm so average: many of my colleagues have an even more diverse portfolio of tools at their disposal. The consequence for employers is that most experienced technical communicators have yet to encounter software we couldn't begin using productively within a day, and become skillful with in about a week. Mastery can certainly take far longer, but most of what we do doesn't require that level of mastery.

Would you rather read well-written documentation, or documentation produced by someone who can make Word 2010 jump up and dance?

To see the flaw in using tool skills as a primary hiring criterion, ask yourself this: would you rather read well-written documentation, or documentation produced by someone who can make Word 2010 jump up and dance? Now ask yourself which of the two skill sets (writing versus formatting) is easier to teach, and you'll know which of the two writers you should hire. All else being equal, which is rare, choose the communicator who also knows your development tools and can use them for layout. And speaking of layout ...

Myth 2: Sans serif fonts are more legible online

"All else being equal," this rule of thumb claims that sans serif typefaces are easier to read on low-resolution displays such as computer monitors, which typically have resolutions of between 72 and 96 dots per inch. This resolution is certainly low, even compared with that of the advanced 24-pin dot matrix printers we abandoned in favor of laser and ink-jet printers, and certainly can't do justice to the fine details of many serif fonts designed for print. In particular, the serifs can disappear entirely, and character outlines may even blur because the variable stroke width that characterises traditional serif fonts lends itself poorly to fixed-size pixels.

Unfortunately, though these assertions all contain a grain of truth, "all else" is almost never equal, and you should distrust any typographic studies that claim otherwise. Many factors can overwhelm the theoretical difference in legibility between serif and

sans serif type, even if we ignore the fact that it's possible to optimise the designs of either typeface style for online display (e.g., "slab" serifs hold up better than thin serifs on screen). The typographic factors that can overwhelm the choice of serif versus sans serif typefaces include:

- How well the typeface itself has been designed to support reading: the broad categories of sans serif and serif type span a tremendous range of typeface legibility.
- Line spacing (aka *leading*): Spacing lines too closely together can make it difficult to read even well-designed typefaces.
- Line width (aka *measure*): Wide lines require greater line spacing to support efficient reading, and the default line-spacing algorithms in most software ignore this need.
- Word and character spacing (aka *kerning* or *tracking*): We're all familiar with the distracting *rivers of white space* that emerge in poorly justified type, but poorly kerned type can be equally problematic.
- Type size: It's easy to forget that you can't simply compare 12 point Times with 12 point Helvetica, given that point size often has little correspondence to the perceived size of the type. Compare Times and Garamond at the same type size and you'll see how much smaller the Garamond type appears.
- The degree of contrast between the type and its background, and the effects of anti-aliasing (i.e., tricking the viewer's eye into seeing pixels where there aren't any in an attempt to improve on a monitor's inherent resolution).

So many other factors influence legibility that a generalisation such as choosing sans serif for online use often leads us to forget that the combination of the abovementioned typographical details is generally far more important than typeface choice *per se*. Since I'm attempting to provoke a reaction in this essay, I'll go so far as to claim that any typeface designed for reading (rather than for display) can be made legible through skilled typography. Karen Schriver has provided a detailed summary of the literature on typographic issues that should be mandatory reading for documentation designers concerned with issues of typography¹.

Even if sans serif fonts were the hands-down winners, enforcing their use ignores perhaps the greatest blessing of publishing information online, something that has thus far been impossible to achieve in print: readers get to choose the fonts and

font sizes that they prefer rather than having to cope with our choices. This is particularly important for visually impaired readers, who will likely become an increasingly significant part of our audience as the average age of our readers increases. My advice, subject again to the caveat that I'm overstating my case to make a point, is that we should never deprive readers of this flexibility unless we've carefully weighed the advantages of what we're giving them in return. In my experience, there's little advantage to enforcing a typeface choice. Using specific typefaces is most important when the graphic design approaches the content in importance, but that's rarely the case for most of the work we do.

One important caution about leaving typeface choice to the reader: make sure you document how readers can make the necessary changes, since many neither know that they have a choice nor understand how to make the change once they do know. But will that change over time? I've said that our audience is ageing, and that's

just the tip of the iceberg.

Myth 3: Audiences are static

There's a myth that once you've characterised your audience through audience analysis, the job's done and all you need to do is follow up with a round of usability testing to provide a reality check. That's far from true.

Inconveniently, audiences insist on changing over time. The neophyte that you devoted an entire *getting started* manual to eventually grows beyond the need for this information, and may even become a power user. Some of the former power users leave, tempted away from the fold by newer, more interesting products that present exciting new possibilities. In particular, the radical fringe who first adopted a product and pushed it to its maximum potential often leave to follow newer waves, leaving behind craftsmen who feel no need for such exploration. And the cycle begins again as more neophytes pick up the product and decide it's worth learning because it's the standard.

I've already mentioned that our audiences are aging, but this has significant implications beyond the need to remember legibility issues. One change that is already well underway and that may be complete within the professional lifetimes of most technical writers involves computer use. Even today, 30 years after personal computers began moving out of the hands of hobbyists and into the hands of regular users, we must write for an audience that includes a fair number of people who are acutely uncomfortable with computers, and who may be using them for the first time. But within one or two decades, these people will have become a vanishingly small component of the audience for

The myth that minimalism equals brevity stems from the view that you shouldn't bury readers in extraneous detail.

1. KA Schriver, *Dynamics in document design*, Wiley Computer Publishing, New York, NY, 1997.

typical software developers. If they become sufficiently rare, perhaps our employers won't grant us the time and resources to cater to their needs. For most of our audience, computers will be so familiar that they're second nature, and that will have profound implications for how and what we document. There's already a trend in this direction, since manuals that begin with the words "We assume you already know how to use Windows" have pretty much driven manuals with an operating system tutorial into extinction.

How else will our audience change over the next two decades? The only way to find out will be to keep our eye on them and start assessing how their needs are changing.

Myth 4: Minimalism means keeping text as short as possible

John Carroll has been one of the leading standard bearers in the minimalism movement, and no doubt has grown rather frustrated with the notion that minimalism means brevity pure and simple. It also doesn't mean trial-and-error learning, maximum simplicity, or any of several other misconceptions or oversimplifications. To set the record straight, he co-wrote an article that deals with the misconceptions firmly and eloquently.¹ Since I lack the space here for a full review of minimalism, I'll risk oversimplification myself by treating the subject in much less depth than it deserves. To quote Carroll: "The central principle in minimalism is task orientation. But many other principles play a role in this design approach either because they support task orientation or because they follow from it". In short, the minimalist philosophy involves understanding what your audience is trying to accomplish (audience and task analysis), and focusing on those needs by providing enough information, in the right form and at the right time or in the right place, to help them accomplish their tasks.

The myth that minimalism equals brevity stems from a much more interesting and complex assertion: that you shouldn't bury readers in extraneous detail. The challenge, of course, lies in discovering what is truly extraneous. It's also a myth that minimalism is a one-size-fits-all solution for all communication problems, since its task orientation does not make it directly applicable to problems such as communicating theoretical information (e.g., the *why* of graphic design rather than the *how*) or writing to

persuade the reader (e.g., marketing writing). Yet even for such seemingly unrelated problems, minimalism has much to teach us because of its emphasis on the reader, and that emphasis won't lead us far astray even when the reader's tasks are not immediately recognisable as tasks. The fact that a philosophy designed for one specific field (task-oriented documentation) can be so easily misinterpreted, yet still have broader implications for communication leads neatly into another myth ...

Myth 5: The optimum number of steps in a procedure is 7 plus or minus 2

George Miller studied, among other things, human short-term memory, but is perhaps most famous for discovering "the magical number seven". Miller's best-known paper² is also probably his least-read paper, and this lack of returning to the source has led to one of the more pernicious misunderstandings in the field of technical communication. In effect, generations of writers have made the assumption that lists and procedures, for example, should

One of the more pernicious misunderstandings in the field of technical communication is that lists and procedures should contain no more than five to nine steps.

contain no more than five to nine steps, based solely so far as I can tell on the title of Miller's paper and the myths that have grown up around it. As it happens, Miller's article actually discusses the human

ability to reliably distinguish categories (e.g., distinct shades of grey or sound levels) and the related issue of *channel capacity*, which represents how much information your audience can manage at one time. In effect, this represents the number of cognitive tools a typical reader can hold in their "mind's hand", so to speak, and use to attack a problem.

I won't try to summarise Miller's rich and moderately dense prose, both because I want to encourage you to read the original article yourself and because an update of this subject merits its own article. Given the importance of what Miller discusses, we should begin thinking about how to test the applicability of this body of research in our own unique context so we can begin applying the new findings to our work. While we wait for those results to trickle in, two things we already know give us much to ponder.

First, we should always go to the source rather than blindly accepting someone else's report of what that source said. This takes longer and usually requires considerably more thought on our part, but it greatly reduces the number of myths and misconceptions that we'll perpetrate. More interestingly, revisiting an article often leads to

1. JM Carroll & H van der Meij, "Ten misconceptions about minimalism", *IEEE Transactions in Professional Communication*, 1996, vol. 39, iss. 2, pp. 72–86.

2. GA Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information", *Psychological Review*, 1956, vol. 63, iss. 2, pp. 81–97.

inspiration and the discovery of new ways to build on old thoughts.

Second, Miller's study does have intriguing implications for technical communication, even if not the ones we've been led to expect. For example, our audiences have very real limits on how much information they can process simultaneously, and recognising the existence of these limits means that we need to better understand how we can help readers to process information. All else being equal, readers will always find it easier to deal with fewer items at a time than many items. As a starting point for applying Miller's findings, we need to learn to write in such a way as to let readers digest one chunk of information before we force them to begin dealing with the next one. And if that means we have to reconsider an interface design because we're asking users to deal with too many inputs at once, then that leads neatly into two more myths ...

Myth 6: You can make a bad interface easy to use through superior documentation

By definition, really good documentation makes even the worst interface easier to use—but it will never make a truly bad product easy to use. I stated earlier that one thing that makes us so valuable to our employers is our ability to think like the product's

users, and if something is difficult to use, we notice it first because we have a devil of a time trying to document how to use it. Our value as communicators lies in our ability to figure out where the barriers to usability lie and create documentation that guides users as painlessly as possible around the problems.

Unfortunately, that's all that most of us have been able to do thus far, and it's time we began making concerted efforts to go one step further. If we can understand the barriers well enough to solve the problems in our documentation, this means we also understand the barriers well enough to propose changes in the interface itself. And we should. Increasingly, that's the role we must take on for ourselves. I'm not the first to recognise this, nor am I the first to propose that we do something about it.¹ But corporate culture is often such that making our voices heard is difficult, and there are many barriers raised in our paths. Why don't we circumvent these barriers? Because of yet another myth ...

Myth 7: We can't talk to the SMEs

This myth arises from the misperception that we can't talk to our subject-matter experts (SMEs) or development teams and persuade them to make necessary changes. Occasional conversations on internet forums have convinced me that some corporate cultures still formally or informally try to prevent technical communicators from bothering SMEs or developers. Apart from making for unpleasant workplaces, this approach can prevent the synergies possible when the two groups collaborate. Fortunately, the truly Dilbertian companies are rare. But even in companies that encourage contacts between us and our partners in product development, it's easy to establish exclusively formal professional relationships that don't foster true collaboration. It's easy to make personal contacts with SMEs, whether at lunch, at the project barbecue, or after hours while awaiting the bus home. These personal contacts are crucial in technical communication because they establish mutual respect and often even affection, thereby earning us the time and open minds we need if we're to get our opinions heard.

Once you've got someone listening, it's relatively easy to keep the conversation going and to start influencing how things get done. After all, a friend is far less likely than a complete stranger to refuse to provide technical information, or to dismiss your concerns out of hand. Even if friendships never develop, the relationships can still become more than merely professional, which means that they generate openings for an exchange of ideas and concessions when it comes to developing and documenting a

Effective Onscreen Editing: new tools for an old profession

Editors are increasingly being asked to edit on the screen using a word processor, but most are finding it challenging to transfer their skills to editing with a word processor. *Effective Onscreen Editing* teaches the basics you need to learn to make the transition, plus proven tips and tricks to maximise your productivity and effectiveness. The book describes general principles valid for any software, then illustrates the principles using Microsoft Word to make them more concrete.

Available as a printed book from Lulu.com and also available as an eBook optimised for onscreen reading.

Learn more at the book's Web page:
<http://www.geoff-hart.com/books/eoe/onscreen-book.htm>

 Diskeuasis
Publishing

1. See, for example, S Carliner, "Future travels of the infowrangler", *Intercom*, Sept.-Oct. 1998, pp. 20-24.

product. For example, although I'm not formally part of the software development team at my workplace, one of our developers now comes to me seeking interface advice, and the others now listen to and consider my advice once we've actually begun working together on the documentation.

Long-term, these sorts of dialogues can subvert even the most toxic corporate culture and produce a relationship in which developers and technical communicators start working together in the interests of our mutual audience. Start building the relationships now. Usability testing is a great way to begin, or would be were it not for yet another myth.

Myth 8: Usability testing is prohibitively expensive and difficult

Any time you try to study human psychology, you're dealing with an inherently complex, subjective endeavour that provides considerable room for error. Usability testing is no exception, and if you're going to perform a refined, statistically sound, replicable usability test, you need to understand a fair bit about both experimental design and human psychology. The good news? Despite these stiff requirements, usability testing need not be prohibitively expensive and logistically difficult, and that makes it easier to justify a series of small, inexpensive tests while you develop the necessary expertise. Jakob Nielsen's experience, for example, has shown that you can get excellent results with surprisingly small test groups.¹ So the fact that so many of us have been scared away from even considering usability testing is unfortunate.

In fact, just about any thoughtfully designed usability test is better than no usability testing whatsoever. As Nielsen trenchantly observed, "[t]he most striking truth ... is that zero users give zero insights". In the worst-case scenario (i.e., you have no resources and you're the only person available to do the work), you can use yourself as a stand-in for your audience, because then you've got at least one data point. I hasten to add that you are not your audience. While you document a product, you'll learn far more about it than most of its users ever will, and in any event, it's hard to imagine how any audience could be less diverse than a single individual. But even when you don't truly represent your audience, many things that you find problematic will pose exactly the same problem for the real audience. If the current interface takes twelve clicks to accomplish a task and you can describe a way to do the task in three clicks, with equal or greater clarity, then you've discovered a more usable alternative. If the software uses six different words to describe the same concept, and only one of the words exists in a standard office

dictionary, then that one common word is often the best word for your audience.

A really useful usability test will require considerably more feedback than you can provide by yourself, and that's where things start getting complicated. Fortunately, a few simple guidelines can let you gather important, helpful, reasonably reliable data from even a small subset of your audience:

- Test your questions and the tests themselves with a colleague, then determine whether you can analyse the answers efficiently and extract useful information. The answers must be easy to understand and summarise, and must let you identify and determine how to resolve the problems they identify.
- Examine your questions and test-designs for signs of bias. Biased questions or designs collect biased answers and can mislead you in your subsequent efforts to improve usability. For example, asking "just how bad is this interface?" gets a very different answer from "how highly do you rate this interface?" because of how the questions focus the respondent's attention. Similarly, testing only a group of experienced users will provide data that won't adequately represent the needs of neophytes.
- Collect data separately for each distinct group in the audience (e.g., neophytes v. experienced users). Programmers who will install and maintain software have distinctly different needs from those who'll use the installed software, and you cannot combine the data for these two groups and still get results that express their different needs.
- Collect data from several people within each distinct group. This reduces the chance of focusing on the one person who is completely different from everyone else in that group. The mathematics of statistics suggest that larger samples are more representative of the overall population, but as Nielsen suggests, "larger" need not inevitably mean "prohibitively large".
- Avoid conditions in the test or the test environment that could influence the results (e.g., the test documentation is written in English but is given to someone with a different birth language, the phone in the participant's office is constantly ringing, their computer is too slow to run the software efficiently, and so on). Each factor that you control introduces some measure of bias, since you'd get different results for situations in which that factor differs. When you try to control a test factor, don't blind yourself to its implications. For example, localised documentation may be required for different linguistic audiences, voice-assisted navigation or spoken online help may prove useless in a noisy

1. J Nielsen, "Why you only need to test with 5 users", *Alertbox*, March 2000, <http://www.useit.com/alertbox/20000319.html>.

environment, and the computer configuration the developers recommended may be unrealistic.

- Confirm that you've understood the answers or results by asking for clarification from the test participant (e.g., "why did you say this product is so bad?").
- Never focus so narrowly on your own objectives or preconceptions that you blind yourself to unforeseen revelations from the test participants.

This brief summary is intended solely to get you thinking about testing. For additional information, I recommend reading more about designing test questions¹, organising and managing the test², and evaluating the results³. Craig provides an annotated bibliography that can help you expand your research.⁴

Myth 9: Single-sourcing means dumping printed documents online

The need to produce online information to accompany printed documentation has been with us as long as software itself. But there's a growing recognition that the interface itself is often the best place for documentation, and that effective online information must integrate effectively with the way people use the product itself.⁵ Complicating matters—as any working professional knows—is the fact that resources are always limiting in the business world, and that there may be insufficient resources to develop information tailored specifically for use in different printed and online information.

The conflict between the ideal of providing information customised for use in a single medium or context and the very real inability to create such information as often as we'd like has generated a tremendous need for single-sourcing: creating one set of information that can be reused for both online and printed documentation. There's a lot of logic in adopting this compromise, since even when resources aren't limited, it makes little sense to duplicate effort by writing the same text twice. Apart from the inefficiency of creating information twice, the risk of introducing discrepancies between the online and print versions of documentation by creating them independently is hard to accept.

Unfortunately, many companies have been misled into simply producing a single set of documentation and dumping it online. (This approach is encouraged by an attitude that documentation is a cost centre, not a benefit to the company and the users of its products. I'll discuss that myth next.) Much of the potential of technologies such as Adobe Acrobat has been wasted by communicators who can't or won't even take the time to reformat a vertical document designed for paper to fit on a horizontal computer screen. At its most innocent, this practice simply shifts the cost of printed documentation to the users of the documentation, who give up trying to read the information online and resentfully print their own copies. At its worst, this form of documentation can be a serious productivity drain. Endlessly scrolling from the top of a page to the bottom or squinting to view a full page displayed in a miniscule font is inefficient and outright annoying to many readers. Indeed, the very word *scrolling* speaks eloquently about how badly this design serves our audience, for if scrolls were such a good communication tool, why then did we abandon them in favour of bound books?

Even when designers have the time to reformat printed documentation for proper display on screen, this approach generally fails to take advantage of the respective strengths of each medium. Despite much compelling writing on the issue⁶, I continue to see online documentation that is unusable or difficult to use online simply because of how it's presented.

There are valid approaches to producing single-source documentation, of which XML is perhaps the best known. But this requires a fundamental shift in the way we look at producing information, both in the initial stages (creation of the information) and in the final stages (presentation or delivery of the information). The failure to understand the distinction between these two phases and the often-forgotten third phase—use of the information by our audience—is one of the main factors behind the current sad state of much online documentation. Yes, you can go a long way towards single-sourcing with appropriate planning and tools; no, you cannot generally pour information from one medium into another without some reworking of the information's content or structure.

Myth 10: Documentation is a cost centre

It's easy to see why technical communicators are often first on the chopping block when it comes time to trim staff: we cost a lot, we make all kinds of unreasonable demands (such as time and money to

1. J Rojek & A Kanerva, "A data-collection strategy for usability tests", *IEEE Transactions on Professional Communication*, 1994, vol. 37, iss. 3, pp. 149–156.
2. L Kantner, "Techniques for managing a usability test", *IEEE Transactions on Professional Communication*, 1994, vol. 37, iss. 3, pp. 143–148.
3. SE Hubbard, "A practical approach to evaluating test results", *IEEE Transactions on Professional Communication*, 1989, vol. 32, iss. 4, pp. 283–288.
4. JS Craig, "Approaches to usability testing and design strategies: an annotated bibliography", *Technical Communication*, 1991, vol. 38, iss. 2, pp. 190–194.
5. C Lockett Zubak, "What is embedded help?", *Intercom*, March 2000, pp. 18–21.

6. See RJ Brockmann, *Writing better computer documentation: from paper to hypertext*, John Wiley and Sons, New York, NY., 1990 and W Horton, *Designing and writing online documentation. Hypermedia for self-supporting products*, John Wiley and Sons, New York, NY., 1994.

perform audience analysis and usability testing), we take developers away from their crucial work to answer naive questions, we hide away in our cubicles and write instead of persuading others to shout our praise in the ears of upper management, and we produce a product that often generates no obvious income for our employer. That's the myth. The facts can be quite different.

There are many ways in which we add value for our employers, including generating both income and savings.¹ The kinds of value we add include:

- Reducing internal corporate communication costs through improved communication.
- Improving the organisation's return on investment.
- Reducing technical support and other post-sale costs.
- Reducing development costs by freeing developers from the painful task of writing.
- Producing saleable materials (e.g., Adobe's *Classroom in a Book* series).
- Reducing the frequency of lawsuits, product recalls and product returns.

It pays to think creatively when you're trying to demonstrate your value. For example, documentation can conceivably generate a sizeable return on investment even in situations where that doesn't seem likely. If documentation is sold as an optional extra for the product, the profits generated by selling it are obvious, but even when the documentation is part of the sale price, there can be tangible profits. Consider, for example, a company with tight accounting controls that sets a target for an overall 20% return on their gross investment. This means that for each \$1.00 the company spends to produce a product (e.g., software), including all expenses incurred by the company, they would earn \$0.20 in net revenue. In such situations, it's easy to overlook the fact that documentation generates part of that \$0.20 profit. Each \$1.00 the company spends on our salaries and benefits, and on printing manuals, also generates \$0.20 in profit for the company! This logic depends on many assumptions, including the assumption that profits are calculated based on total expenses rather than purely on development costs, as might be the case in a startup company. It also assumes that the company is sufficiently well run to generate a profit, let alone one as healthy as 20%. But the logic still applies to some

companies, and it should be possible to obtain the data you need to determine whether it applies to your own company and generate some real numbers that demonstrate your net worth to your employer.

I find this myth particularly interesting because it's both held about us by others and something that we believe about ourselves. We each intuitively understand the value we add—even if we're not able

to easily articulate that value—but because we haven't taken the time to demonstrate that value to others, we begin to doubt our own value. It's past time we began changing

management perceptions so that they understand our true value, whether we measure that value in tangible or intangible ways. Do some of the work necessary to define this value, and management will begin taking you more seriously. That's a good first step towards feeling more secure about yourself, your value to your employer and your job security.

Conclusion

Myths aren't always invalid. They endure because no matter how much they simplify or exaggerate reality, they are nonetheless based on something truthful, something important to us or something that sheds a bright light on an aspect of our lives. One of the things that fascinates me most about mythology is just how universal the themes can be, and how creatively each person or culture can be in reinventing a myth by recasting it in their own unique context. Folklorist Josepha Sherman has observed that "Myths are attempts to explain the cosmic truths ... All peoples have the same questions, and so all peoples have the same basic type of myths". Each of the ten myths I've presented in this paper passes this test for that idiosyncratic group of people known as technical communicators. My hope is that each of us will find ways to answer those universal questions for ourselves by seeking out the underlying truths and building on them to create something more useful and fascinating still. By making the myths more relevant to us, we reinvigorate them and ourselves. One obvious way to do this is to re-examine our current rules of thumb and see how they can be refined. After all, the thing to remember about rules of thumb is that thumbs bend when the situation calls for it.

Geoff Hart

Geoff Hart is a scientific editor who specialises in working with authors for whom English is a second or third language. He is also the author of *Effective Onscreen Editing* (<http://www.geoff-hart.com/books/eoe/onscreen-book.htm>). Geoff also works as a French translator and occasionally as a technical writer. In his spare time, he commits bloggery (<http://blatherskite.dreamwidth.org/>).

This article is an updated version of an article of the same name published in 2000 in *Technical Communication*, vol. 47, iss. 2, pp. 291–298.

There are many ways in which we add value for our employers, including generating both income and savings.

1. See JG Redish & JA Ramey, "Measuring the value added by professional technical communicators", *Technical Communication*, 1995, vol. 42, iss. 1, pp. 23–83 and J Mead, "Measuring the value added by technical communication: a review of research and practice", *Technical Communication*, 1998, vol. 45, iss. 3, pp. 353–379.

Font choice revisited

Geoffrey Marnell

In an earlier issue of *Words* (vol. 1, iss. 2) I summarised the results of the research that Colin Wheildon undertook on, among other things, the relative comprehensibility of text set in a sans serif font (such as Helvetica) and the same text set in a serif font (such as Corona). Wheildon’s research backed up what publishers have suspected for a very long time: that body text set in a serif font is more comprehensible than body text set in a sans serif font. (Open at random any book intended for adult readers and note the font used for the body of the text. London to a brick, it will be set in a serif font.)

Table 1: Comprehension scores after reading serif and sans serif body text			
	Percentage of correct answers given		
	70–100%	40–69%	0–39%
Serif font	67	19	14
Sans serif font	12	23	65

Let me republish a table from that earlier issue of *Words* to re-emphasise the strength of the effect Wheildon observed (see table 1 above). By way of explanation: of those testees who were asked to read text set in a serif font and then answer questions about what they had read, 67% got a score of 70% or more. That is, they answered at least 7 out of 10 questions correctly. On the other hand, of those testees who were asked to read the same texts but set in a sans serif font, only 12% got a score of 70% or more when asked the same questions. On those figures, serifs win hands down.

The academic literature on this topic is sparse. Earlier this year I was alerted to a 2005 literature review entitled “Which Are More Legible: Serif or Sans Serif Typefaces?”

undertaken by the graphic artist Alex Poole.¹ The review almost exclusively deals with readability as it applies to graphic design (legibility, line measure, white space, leading and so on) and not to readability as it applies to language (comprehension, understanding, recall and so on) and thus is concentrating on matters other than those Wheildon concentrated on. But Poole does make one brief mention of Wheildon’s work:

“There are some high profile studies which claim to show the superiority of serif typefaces (Robinson *et al.*, 1983; Burt, 1959; Wheildon, 1995) but these have been soundly criticised on points of methodology. (Lund, 1997, 1998, 1999).”

If we are writing instructions to assist those in the control room of a nuclear power plant handle emergencies, our choice of fonts may well be the difference between life and death.

What is odd, and somewhat discouraging, about this statement is that in a review of, on Poole’s admission, “over 50 empirical studies” he could find only one person critical of Wheildon’s work: Ole Lund. Surely if Wheildon’s quite extraordinary research had been thoroughly debunked, there would be much more comment in the literature.

Lund’s review

Of the three references Poole gives to Lund’s work, only one explicitly mentions Wheildon in the title: a three-page review of Wheildon’s book *Type & layout: Are you communicating or just making pretty shapes* published in 1998.² This is such an extraordinary review that it is worth dissecting its six short columns column by column.

It is a case study in how *not* to write a review and expect it to be seen as carrying any weight.

- Column 1 and half of column 2: Lund gives the history behind Wheildon’s publication and a summary of his major result, without analysis or criticism. So far, this is fine, despite the abundance of snigger quotes and the claim that Wheildon’s results are “sensational”, a claim that is hardly neutral.
- Rest of column 2 and all of column 3: Lund takes Wheildon to task for explaining the results of his research with a “rather dubious appeal to the authority of past research into [the optical phenomenon of] irradiation”. But an appeal, dubious or otherwise, to past research is not an argument against the existence of the phenomenon one is trying to explain. Serifs might still provide greater comprehension irrespective of whether irradiation is the reason, just as the earth might still be warming even if solar activity is found not to be the cause.
- Column 4 and most of column 5: Lund quotes a number of the testimonials—from graphic designers, editors and academics—that Wheildon includes in his book and calls it “hilarious” that so many have been included. But even if Wheildon did go over the top in a “persistent appeal for external praise and legitimacy” (as Lund puts it), this is simply not relevant to

1. My thanks to Stuart Burnfield for alerting me to this review. It can be read at <http://www.alexpoole.info/academic/literaturereview.html>.

2. Lund’s review can be found in *Information design journal*, 1998, vol. 9, iss. 1, pp. 74–77. My thanks to Dr Bill Parker for tracking down a copy of this review.

whether Wheildon's methods and conclusions are sound or unsound.

The attack on Wheildon, not on his research, starts to get nasty when, of the people Wheildon claims to have provided him with advice during his research—many of whom are or were academics—Lund asserts: “My guess is that it is doubtful that these persons were involved in Wheildon's work in the way that he implies, and that they would not authorise such use of their names”. A guess? This is hardly convincing. Indeed, it is a case of *argumentum ad hominem*: attacking the man and not the argument. Even if Wheildon did exaggerate the intellectual influence of others, how is that at all relevant to the issue of font superiority?

- Rest of column 5 and all of column 6: Lund lists two works on information design where Wheildon's results are cited (books by Kempson and Moore, and by Karen Schriver) and he calls them to task for accepting the “blatant outrageousness” of Wheildon's results “more or less uncritically”. Uncritically? But where is Lund's logic-based or fact-based criticism?

Lund criticises Wheildon's conclusion but does not provide one skerrick of evidence to contradict it: his own or anyone else's. He talks of Wheildon's “findings” and his “empirical evidence” — snigger quotes included — as if Wheildon never engaged in a study or drew findings from a study. The whole review is an blatant *argumentum ad hominem*, suited to the opinion pages of a newspaper, not to an academic journal.

Lund's article

Poole gives two other references—again written by Lund—that supposedly “soundly criticised on points of methodology” the work of those who thought they had established the superiority of serif fonts. The 1997 reference is to an article titled “Why serifs are (still) important”.¹ (Yes, the distracting, non-sensical parentheses are in the title.) But in this paper, Lund entirely focuses on research reported in 1971 by David Robinson and colleagues.² There is not a single mention of Wheildon, neither in the body of the paper nor in the references.

Lund's thesis

The third reference Poole gives is to Lund's 1999 PhD thesis, entitled *Knowledge Construction in Typography*. The thesis is neither published nor generally available, and thus has not had the benefit of general

scrutiny. For that reason, it simply does not deserve to be listed in a literature survey. It may well make some good points against Wheildon, but who can easily tell? I have sent two emails to Lund asking for access to his doctoral material relating to Wheildon. Neither has been answered. It is difficult to comprehend why research that supposedly counters a longstanding publishing practice, and Wheildon's own research, remains unpublished and seemingly guarded by its author. Most academics are more than willing to argue about their views—unless they have abandoned those views.

What does it all mean?

So, *pace* Poole, Wheildon's work has not been “soundly criticised on points of methodology” by Lund. A vitriolic, substance-less book review does not constitute sound criticism. Nor does a work untested by general peer review. Wheildon's results may well be wrong. But more research—serious research—needs to be done to prove that point (or to back up Wheildon).

This is a critical issue. It should interest all technical writers. If we want our readers to maximally comprehend what we write, then font choice may well be a critical decision we make. If we are writing instructions to assist those in the control room of a nuclear power plant handle an emergency, our choice of font may well be the difference between life and death.

Other reasons for preferring serifs

Even if serifs and sans serifs were equally legible and equally comprehensible, there may be other reasons to choose a serif font over a sans serif for body text. For instance, the width of most serif characters is narrower than the width of corresponding sans serif characters. A 100-page document set entirely in Arial would become just 83 pages if it were set in Adobe Caslon Pro (or 91 pages if set in Times New Roman).

This calculation assumes that the characters in the alphabet are used equally throughout the document. This might not be the case. But even if we restrict the comparison to, say, the 13 most frequently used characters in written English—*e, i, s, a, r, n, t, o, l, c, d, u* and *g*—Adobe Caslon Pro is still more economical than Arial: 90 pages for every 100. The same result applies with Times New Roman. Garamond Premier Pro is even more economical: just 81 pages for every 100 pages of Arial. The sans serif font Calibri is better than Arial. Nonetheless, 100 pages of Calibri is the equivalent of just 91 pages of Garamond Premier Pro.

Given that most of us print documents found online if they are more than a page or so long, choosing a serif font for body text may well be the environmentally superior, not to mention the financially superior, choice.

Geoffrey Marnell

1. Ole Lund, “Why serifs are (still) important”, *Typography Papers*, iss. 2, 1997, pp. 91–104. My thanks to Diane Dilby at the University of Reading for providing me with a copy of Lund's paper from this now-out-of-print journal.

2. DO Robinson, M Abbomonte and SH Evans, “Why serifs are important: the perception of small print”, *Visible Language*, vol. 5, no 4, pp. 353–59.

The changing world of product documentation

Dan Smith

Over the years, the positioning by organisations of their product documentation has changed, along with the role of the documentation areas. This has influenced the technical writer role. A symptom of this change is that, in the northern hemisphere at least, technical writer pay rates have been on a downwards trend for the past few years.

The benefit of quality documentation to organisations seems to have become less obvious as time passes. Increasingly, documentation is seen purely as a cost area, necessary only to meet basic user-information obligations.

In my opinion, the following have contributed to this situation:

■ The World Wide Web

Up until the early- to mid-noughties, internet search engines were inefficient at finding information, and there was far less content available. In its early days, Yahoo used flesh-and-blood people to categorise new web pages for their search engine. In those times, product documentation was the only source of information available to most users. Nowadays users often resort to an internet search engine as their first point of reference before looking—if they look at all—at the product documentation.

■ Improved user sophistication

Until recently, computers were not widely available. Many a technical writer's audience only saw a computer at their place of work. Nowadays, many of our potential readers cannot imagine a world without computers. Computers are as ubiquitous as cars and telephones. An increasing number of users have grown up with the internet. They are comfortable with the idea of exploring the functionality of an application they know nothing about.

Users are also more adept at extracting the information they need from disparate web sources. They are adept at browsing and at opening links in new tabs so they don't get lost. They are more able to extract the information they need from content spread over several sites. Moreover, from a diet of blogs, social networking sites and instant messaging, they are comfortable with inconsistent content. Hence they are more prepared to learn by exploring and asking, and will only seek information from the product documentation when they come to a dead end.

■ Improved user interface design

As time passes, user interface usability improves and design principles converge across applications. Users know what to expect from a

new application before they start, and they have a good idea of where to begin without looking at product documentation. After all, they go through a similar process whenever they buy a new computer game or open a new web site.

Organisations see this changing world as a great opportunity to save money by downsizing their product documentation areas. They can achieve this in a number of ways.

■ Self-documenting, and even self-localising documentation

With this approach, basic product documentation is delivered over the web in a wiki-type mechanism. Users can edit existing content and add new content, and even localise the content. The organisation retains a minimal writer presence to vet and sanity-check changes. This is an emerging technology that is yet to be put to the test.

■ Outsource documentation preparation to the cheapest supplier

The cheapest suppliers are usually in countries where English is not the native language. Up until recently these suppliers had a poor reputation for quality, but emerging technologies similar to localisation applications allow writers to select appropriate phrases and content from a mechanism similar to translation memory. These technologies can also apply a basic quality control and consistency check.

■ Transferring resources from product documentation to product support

At the expense of documentation areas, organisations increase resources in their customer support areas to ensure that response and issue resolution times are satisfactory. Whereas product documentation is seen as purely a cost area, customer support can generate revenue through support contracts. The icing on the cake for an organisation's bottom line is that the take-up rate of support contracts can be improved by supplying inadequate documentation.

■ Promoting and supporting product user groups

Organisations set up and actively promote user groups for their products. Users cooperate to resolve problems and share product knowledge. Organisations ensure that these user groups are monitored by product experts so that the user group's credibility and popularity remains high. Again, the success of these can be improved by inadequate product documentation.

For technical writers to survive and thrive in this environment some new strategies are required.

It is more important than ever to become an expert in the products whose documentation you are responsible for. As well as improving the quality of your work (not that quality matters so much in the new world order), it allows one the opportunity to be proactive in product design and documentation delivery. It is important to be taken seriously as a product expert when it comes to contributing to matters like user interface design.

It is also important to get to know the customers and users so that you can spot opportunities for new ways for supplying them with the information they need. Ideally, your efforts will affect customer support and, with luck, product marketing. Any visibility that you can achieve in the product marketing area is priceless. As a writer, you need to diversify more. You can, for example, select an additional area of expertise and focus on gathering some serious skills in that area.

For the non-technical technical writer, the extra-curricular areas to focus on are user interface design, business analysis, product marketing and product training. Business analysis especially is worth looking at, since there is much overlap with technical writing and task analysis, and there is a demand.

For those of a more technical bent, there are many options available. As a product matures, the amount of related information increases exponentially. The ability to program in scripting languages is very useful for handling information in bulk. For example, scripts can save serious time when changing product names or implementing company branding changes across a large product documentation set.

Knowledge of JavaScript and skills in creating cascading stylesheets (.css files) are extremely useful for setting up the look of, and implementing functionality on, the organisations's document site. If your organisation uses XML or even HTML, then extensible stylesheet language (XSL) is a very powerful tool, and if you can get your head around it, XSL skills are very much in demand. The things you can do with XSL are truly surprising.

To sum up, we live in a changing world. So what's new? The key is to look for the opportunities that these changes bring.

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.



MACQUARIE DICTIONARY ONLINE

www.macquariedictionary.com.au

Subscribe to the complete Macquarie Dictionary online, updated annually with new words and definitions.

Also available online is the full Macquarie Thesaurus - that perfect word is just a click away.

Try it out now for FREE!

Macquarie Online is offering free extended trial access. Simply contact Macquarie Online to set up your 3 months free access. Quote code: 3mfTrialAC

Macquarie Online Support
phone: 1800 645 349
email: support@macquarieonline.com.au



Australia's national dictionary

commercial
translation
centre

**For Effective
Translation/Localisation
Of Technical Documentation
In 80 Languages**

Australia Wide Service Since 1988
Call 1800 655 224 (Australia Toll Free)
Email: mail@ctc.com.au
Web: www.ctc4.com

commercial translation centre

Numbering in FrameMaker 9 books

Mark Ward

In FrameMaker 9 you can create hierarchical books using folders and groups. This adds to the ways in which you can organise the content of a FrameMaker book. Note that adding a folder or group to a book does not add a folder to your system. This is unlike RoboHelp, where adding a new folder to your project from the **Project Manager** pod also adds a new folder to your system, one visible in Windows Explorer.

FrameMaker 9 also offers more numbering options than previous versions. For example, there are numbering options for Asian languages (see figure 1) and numbering options for sections and sub-sections.

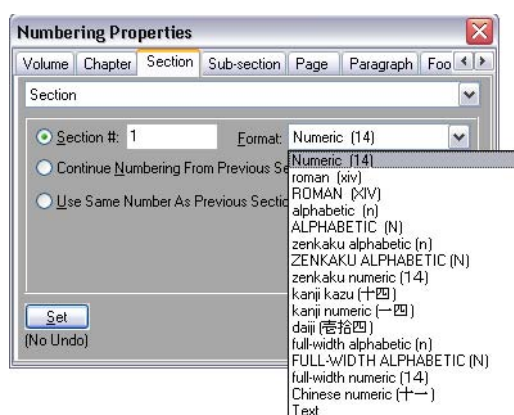


Figure 1: New Asian language numbering options

This article describes how section and sub-section numbering works in FrameMaker 9 books (along with chapter numbering).

Groups and folders

The following graphic shows a book with a group (**Admin**) and a sub-group (**Tasks**).

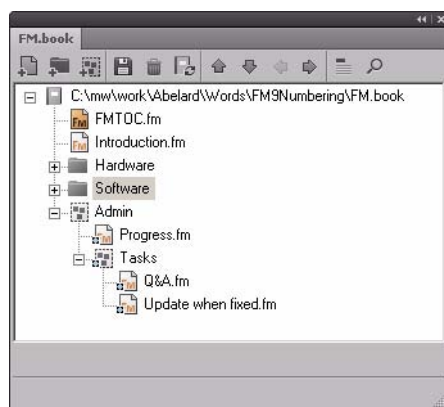


Figure 2: Groups and sub-groups

Numbering properties can be set for groups and sub-groups *with the exception of section and sub-section*. If you select a group or sub-group from a book file and then open the **Numbering Properties** dialog box,

the **Section** and **Sub-section** tabs are disabled.¹ All the other numbering options are available (such as volume, chapter, page and so on). This is also the case if you open the **Numbering Properties** dialog—**Format > Document > Numbering**—from a file that is not in a folder.

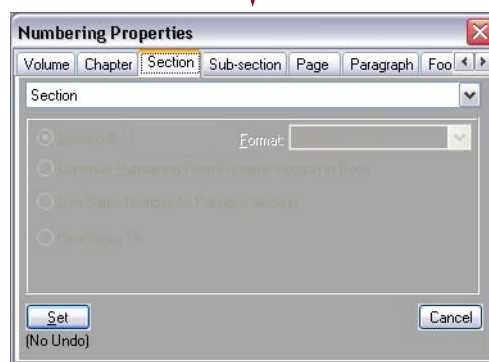
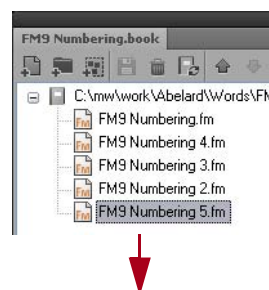


Figure 3: Inactive **Section** and **Sub-section** tabs

If you open the **Numbering Properties** dialog from a group file or sub-group file rather than from a book, the **Section** and **Sub-section** tabs are enabled, but any settings you make on them have no effect in the book.

Section numbering works with files that are in a FrameMaker book *folder*. Sub-section numbering works with files that are in a FrameMaker book *sub-folder*.

An example

Figure 4 (on the next page) shows a book with folders and sub-folders. In this book, chapter, section and sub-section numbering were all set to continue from the previous chapter, section and sub-section respectively.

The Words team

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

© Individual contributors or Abelard Consulting Pty Ltd, 2010
[unless otherwise noted]

1. You can also open the **Numbering Properties** dialog by right-clicking the group or sub-group and selecting **Numbering**.

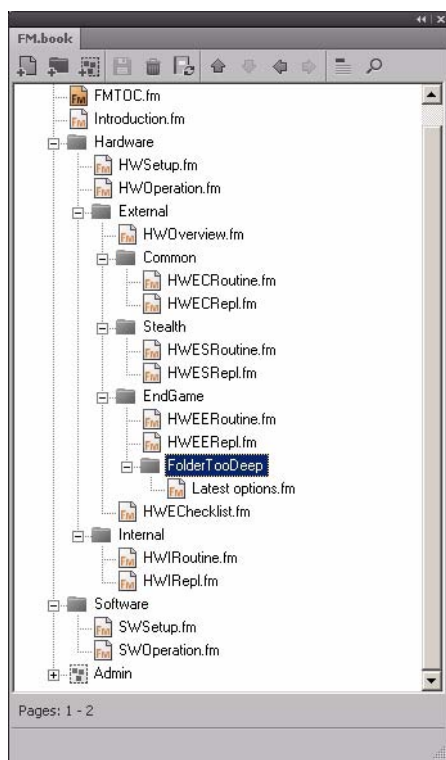


Figure 4: Sample book file

Let's consider just chapter, section and sub-section numbering. As for chapter numbering, it is available at the top level of the hierarchy: for the **Introduction** file, and for the **Hardware** and **Software** folders. Section and sub-section numbering is inactive for files and folders at this level.

At the second level in the hierarchy, section numbering is available, but not chapter or sub-section numbering. In figure 4 above, this applies to:

- HWSetup.fm
- HWOOperation.fm
- External [folder]
- Internal [folder]
- SWSetup.fm
- SWOOperation.fm

At the next level, sub-section numbering is available, but not chapter or section numbering. In figure 4, this applies to:

- HWOOverview.fm
- Common [folder]
- Stealth [folder]
- EndGame [folder]

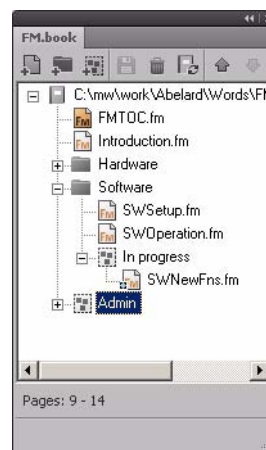
Note that chapter, section and sub-section numbering are inactive for the files in the three folders just mentioned, and for folders and files at lower levels in the hierarchy. This applies to the

folders and files at or below the level of the **FolderTooDeep** folder. These simply take on the numbering of their parent folder.

Note also that if you set the numbering for all folders and files to **Continue Numbering from Previous File in Book**, section and sub-section numbering restarts from 1 with each folder. This is illustrated below:

Document	Chapter	Section	Sub-section
Introduction.fm	1	1	1
HWSetup.fm	2	1	1
HWOOperation.fm	2	2	1
HWOOverview.fm	2	3	1
HWECRoutine.fm	2	3	2
HWECRepl.fm	2	3	2
HWESRoutine.fm	2	3	3
HWESRepl.fm	2	3	3
HWEERoutine.fm	2	3	4
HWEERepl.fm	2	3	4
Latest options.fm	2	3	4
HWECRepl.fm	2	3	5
HWIRoutine.fm	2	4	1
HWIRRepl.fm	2	4	2
SWSetup.fm	3	1	1
SWOOperation.fm	3	2	1

This numbering scheme applies even if a file is in a *group* inside a folder. In the example shown below, the numbering works as though the files are part of the **Software** folder.



Document	Chapter	Section	Sub-section
HWIRRepl.fm	2	4	2
SWSetup.fm	3	1	1
SWOOperation.fm	3	2	1
SWNewFns.fm	3	3	1

Mark Ward

Mark Ward is an independent documentation consultant, adviser to Abelard Consulting, and delivers Abelard Consulting's *Adobe FrameMaker* and *Structured Authoring with Adobe FrameMaker* training courses.

Journals

Journal of Technical Writing and Communication

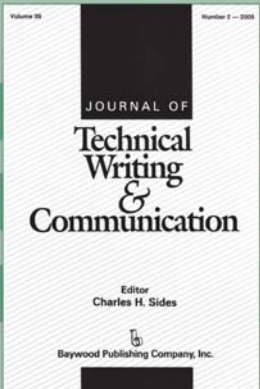
Contents of the current issue

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

Volume 40, Issue 4, 2010

- Linguistics from the Perspective of the Theory of Models in Empirical Sciences: From Formal to Corpus Linguistics, Teresa Grabinska and Dorota Zielinska
- The Impact of NSF and NIH Websites on Researcher Ethics, Ryan Hoover
- Interpretive Discourse and Other Models from Communication Studies: Expanding the Values of Technical Communication, Sean D. Williams
- I, Pronoun: A Study of Formality in Online Content, Alexander Thayer, Mary B. Evans, Alicia A. McBride, Matt Queen, Jan H. Spyridakis
- Resisting the Lure of Technology-Driven Design: Pedagogical Approaches to Visual Communication, Kathryn M. Northcut and Eva R. Brumberger
- The Anatomy of a Book Review, James Hartley
- Book reviews, Elizabeth Tebeaux

award-winning, authoritative international voice, publishing the latest research by recognized scholars from around the globe



Every article ever published in the JTWC is now available in clear, concise, comprehensive PDF format.

New Online-Only Delivery Options

- Enhanced subscription (all articles from volume 1 through current volume)
- Current volume subscription
- Individual articles on pay-per-view basis

Click here for all the details and ordering information.

- Read all article abstracts free
- Download a complimentary sample issue

BAYWOOD PUBLISHING COMPANY, INC.
<http://baywood.com>

HyperTrain dot Com

HyperTrain dot Com specialises in:

- Online Help
- HTML
- JavaScript
- CSS
- XML
- XSLT and
- DITA.

We also offer a full range of tech pubs services including group or individual training, project jumpstart, technical writing, and end-to-end project development.

Toll free (US): 888-722-0700
International: +1-760-214-0698
Web: www.hypertrain.com
Email: dgash@hypertrain.com

AUSTRALIAN STYLE

A NATIONAL BULLETIN ON ISSUES IN AUSTRALIAN STYLE AND ENGLISH IN AUSTRALIA

Announcing the online rebirth of the biannual newsletter published at Macquarie University, featuring:

- o articles on language usage and research
- o the Feedback questionnaire
- o the exclusive crossword Rubicon by David Astle
- o cartoons by Judy Dunn
- o book reviews, a new word column, and much more

Website: http://www.ling.mq.edu.au/news/australian_style.htm

Contact: adam.smith@mq.edu.au

Subscribe to Words

Words is a free, quarterly bulletin for technical writers and communicators—indeed, for anyone with a passion for words and for what you can do with them. To subscribe, send an email to:

words@abelard.com.au

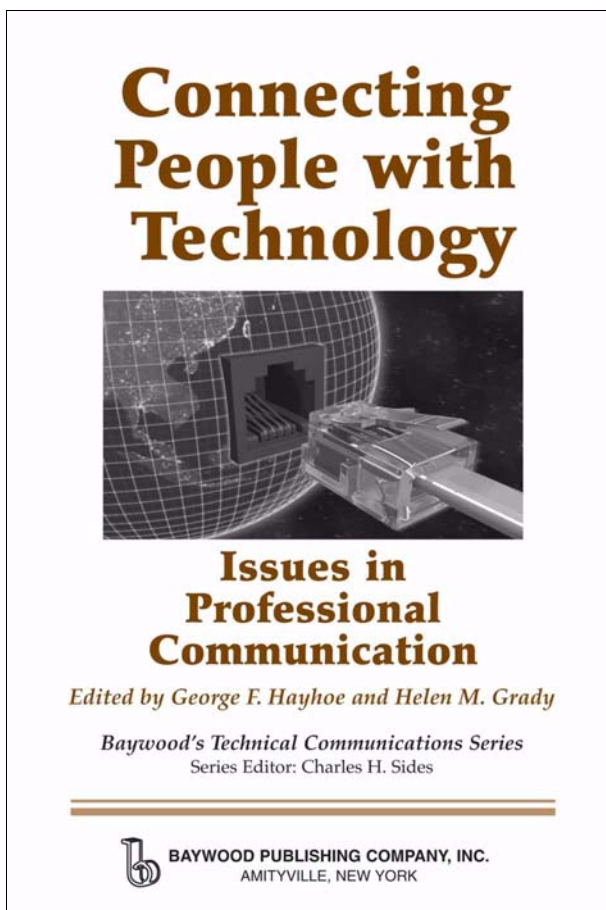
Book reviews

- *Connecting people with technology: Issues in professional communication*, edited by George F. Hayhoe and Helen M. Grady, Baywood Publishing Company, 2009, 280 pages, US\$56.95

Reviewed by Jill Nicholson

This book is a series of articles that evolved from the 2005 IEEE International Professional Communication Conference. The editors chose topics based on their importance to the technical communication profession. The topics are usability, globalisation, health and safety, biotechnology and corporate communication. While the emphasis on these aspects of our profession may differ slightly in Australia—because of our position in the economic world and our geographic isolation—there is something here for everyone. Each author is an acknowledged expert in their field and the editors believed that the author's insights, presented at the conference, were worthy of a wider audience.

I found this book interesting and not so interesting. For a teacher it is a wonderful source of case histories; for a writer it is a more limited experience. The not-so-interesting was mainly because I do not operate in the health and safety or biotechnology fields. There were several articles on these topics throughout the book.



Part I—Usability: Making Technology Fit its Users

Chapter 1, *Making Connections: Teaming Up to Connect Users, Developers, and Usability Experts*, is based on a usability test for a new server development. It is well documented and covers a lot of issues faced by small IT ventures. How often, for example, is there no mental model, issues with navigation and feedback, and an overly complex workflow? The findings in this study are based on a real case scenario yet have a much wider application. It is an excellent article.

Chapter 2 discusses usability standards. Chapter 3 deals with navigation through a website, and Chapter 4 considers text characteristics that help or hinder the elderly. The web-site navigation case study was knee replacement and I found this particularly interesting.

Part II—Globalisation [sic] Overcoming the Challenges of Language and Culture

In Chapter 5, *Communication is the Key to Global Business*, Richard Schaler argues the case that communication is the “tie that binds” our global community. Chapter 6 addresses the issue of hidden costs in localisation. Chapter 7 is about saving money by connecting the writing process to the update and translation process. Chapter 8 addresses the problem of cross-cultural miscommunication and the outsourcing of writing. The final chapter is an assessment of the advice available to non-English speakers who need to present in English.

All these articles are based on well-accepted ideas, but the content is now less novel than it was when they were written.

Part III—Health and Safety: Informing Society of Risks and Dangers

The chapters in this part relate to public discourse in relationship to terrorism, effective communication within humanitarian relief organisations (particularly during relief situations), persuading visitors to visit safe sex sites and the building of relationships between physicians and patients. All chapters are specialised studies and are very academic in attitude. However, from a general perspective they are indicative of the growth of new audiences and new subject domains.

Part IV—Biotechnology: Reporting its Potential and its Problems

This part extends the general perception of growth that I found in Part III. The chapters are all related to extending the communication influence, written and oral, into biotechnology. They include connecting popular culture and science, miscommunication with the public applied to biotechnology based on rhetorical assumptions, and the need for technical communicators to facilitate negotiations in controversial technology transfer cases. I believe that

the principles outlined in these papers can be applied to the development of specialised communication in any science.

Part V—Corporate Environment

Technical communicators often work in complex, technologically sophisticated environments. All six chapters of this final section address aspects of communicating in these environments.

Chapter 17, *Technical Language: Learning from the Columbia and Challenger Reports*, analyses the consequences of taking technical writing as more definite, unequivocal and agreed upon than it really is. This is a superb case study. The author states, “Through these reports we can see in detail how language—in writing, speaking, emailing, and even thinking—influenced these technological endeavours”. Both disasters provided excellent opportunities to learn vitally important lessons about how technical and professional communication occurs in complex and technologically sophisticated organisational contexts. This was the most interesting chapter of the book.

Chapter 18, *The Theoretical Foundations of Service Leadership: A New Paradigm*, is primarily aimed at leadership in a service industry, but does provide good information for anyone interested in the concept of leadership. This is a theory-based paper, but it also relates to the importance of communication at all stages of the defined process.

Chapter 19, *Managing Collaboration: Adding Communication and Documentation Environment to a Product Development Cycle*, discusses a common practice but a new model. The case study was based on the use of communication and documentation models to produce e-learning games for a university where there was a high turnover of multidisciplinary specialists.

Chapter 20 deals with managing international virtual teams, and Chapters 21 and 22 with content management. Chapter 21 deals with the problems of information management in a global industry as large and complex as aerospace. The last paper in the book, Chapter 22, is a study of three cases to make tacit knowledge visible in web-based surroundings. The three cases were developed at the National Institute for Health and Environment in the Netherlands.

This book is a great compilation for readers of technical communication books: there is something for everyone. However, as I said at the beginning, I believe the real value in this book is as a teaching resource.

Jill Nicholson

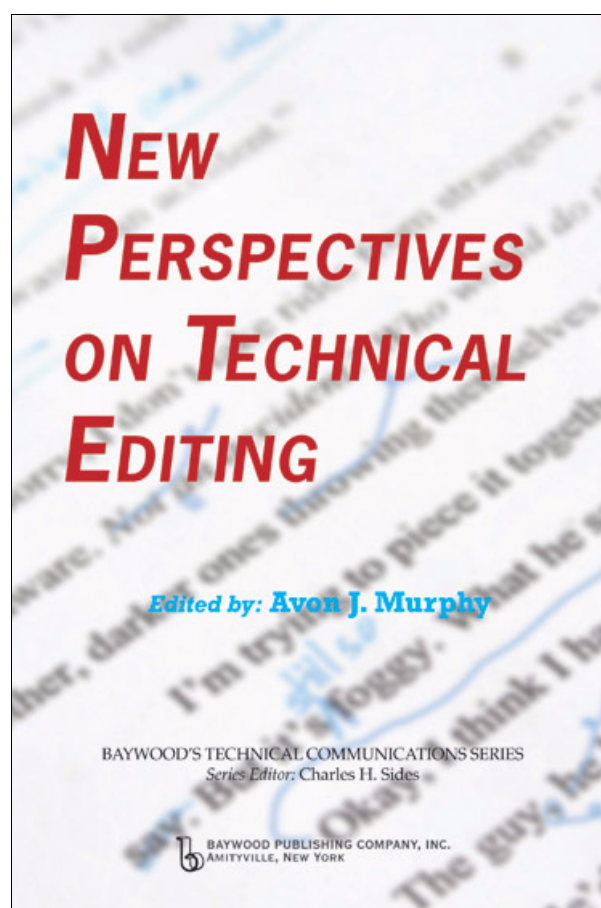
Jill Nicholson founded the documentation consultancy *N&H Communications*. As well as developing and editing documentation, Jill runs training courses in technical writing and tender writing.

■ *New perspectives on technical editing*, edited by Avon J Murphy, Baywood Publishing Company, 2010, 214 pages, US\$46.95

Reviewed by Diane Josey

Being somewhat new to the field of editing and actively interested in the profession of technical editing, I was keen to read this volume of nine essays to find out exactly what defines a technical editor and makes that kind of editor so distinct from other kinds of editors.

On page one of the introduction, Murphy states that, for the purposes of the book, technical editing will be taken to refer to “the planning, analysis, restructuring and language changes made to other people’s technological or scientific documents in order to make them more useful...”. The intended audience for the book is “teachers and students in upper-division and graduate technical communication courses, researchers, and practicing editors...” and the editor notes that “we do not target readers looking for step-by-step tutorials on daily editing procedures”. Murphy believes that technical editing is a “maturing discipline”, one that can profit by a degree of critical self-scrutiny. The reader is thus led to expect, and largely receives, an intelligent discussion of where technical editing currently sits, of the problems confronted and overcome by its practitioners, and of where it might be headed in the future.



Each of the eight writers represented in the book is an authority in their own field and is highly respected in the world of technical editors.

Angela Eaton, the author of Chapter 2, *Conducting Research in Technical Editing*, and Thomas L. Warren, author of Chapter 3, *History and Trends in Technical Editing*, both argue the case for a move in teaching the discipline of technical editing from an historic, anecdotal, localised methodology —based on a model of received advice and observation from a particular practitioner—to one grounded in research. Eaton is intent on preaching to the profession, in a very informed way, the need for a research base and indicates ways in which this might best be achieved. There is a united positivism in the voices of these two authors in the way they urge that a professional approach to technical editing be taken by students in, and practitioners of, the area.

Warren contextualises the profession as practised in the past and in the present. His chapter is particularly learned, thoroughly researched and likely to be of special interest to those wanting to know more about the historical development of technical editing and of its continuing evolution. (I particularly enjoyed his account of the twelfth century editorial interactions between Hildegard and Volmar.) He sees editing as evolving rapidly: from the original printer's checker to the modern readers' advocate and beyond. In favouring the continued "professionalising" of the area—and to emphasise its primary goal of producing audience-directed, usable text—he urges a strong forging of links with other professions, such as psychology (with its emphasis on cognitive understanding). He advocates developing the writer–editor relationship as fully as possible and believes that to attain that, the editor must gain a deep understanding of memory and of other cognitive processes, a point reiterated by others in this book.

In her article, *Copyediting and Beyond*, Jean Hollis Weber also runs with the idea of change, clarifying just what copy-editors do today that they did not do in the past. In a matter-of-fact, knowledgeable, style, she argues the case that there is more to modern copy-editing than merely correcting text. Today's copy-editors need to move with the times and familiarise themselves with electronic deliverables, such as wikis and interactive tutorials. (Is the implication that thereby they become technical editors?) She includes in the copy-editor's skill list the ability to improve textual organisation, readability, retrievability and, judging by the emphasis accorded them in the previous chapters, audience suitability and easy usage.

There are two chapters unlike the others in terms of the specificity of their subject matter: Caroline Rude's somewhat directionless chapter on the teaching of editing, and George Hayhoe's chapter titled *Editing a Technical Journal*. The intended audience of Rude's chapter is unclear. But Hayhoe's chapter will be of specific interest to one obvious sector of the editing world: the editor working on large-scale technical journal publications. It is a thorough, process-oriented piece of writing, delineating approaches to specific problems such as editing texts by non-native English speakers.

In a review such as this, it is impossible to do justice to the thoughts and writing of each of the authors. Perhaps it suffices to say of three of the others, Geoffrey Hart, Barbara Gastel and Michelle Corbin, that they represent themselves as ultimate professionals via their informed, thorough, clearly developed and very user-oriented writings.

Hart clearly places developmental, substantive, copy-editing and proofreading in context and in relationship with one another. He provides comprehensive overviews of the tools of the trade, such as essential software, communication and research tools. He discusses the place of web-based programs and the implications and challenges of single-sourcing and information dependency. He also, as many of the other contributors did, refers to

the essential soft skills editors require in order to develop successful relationships with other stakeholders.

Barbara Gastel's chapter, titled *Editing in the Pure Sciences*, is a must-read for every technical editor,

scientific or not. It provides a wealth of information and advice. Her perception of technical editors sees them as both editors of technical and scientific texts and proficient users of modern technology. Her reference list is vast, up-to-date and invaluable.

Michelle Corbin's *The Editor within the Modern Organisation* provides an insightful peep into a technical editor's roles and responsibilities, including the obstacles and possibilities of such roles. She provides lists of hard and soft skills needed by the technical editor. In providing three scenarios into which editors might fit within an organisation, she presents readers with variations on her own experiences from which they can extrapolate understandings to suit their own environments. In confronting the evolving nature of the technical editor, and discussing modern developments such as information architecture, she reminds the reader that the technical editor remains firmly grounded in "traditional publishing roots".

In most ways, the book is a great success in helping the reader to understand the discipline. The writers discussed above are mostly superb; the

"[This book provides] an intelligent discussion of where technical editing currently sits, of the problems confronted and overcome by its practitioners, and of where it might be headed in the future."

annotated bibliography is comprehensive, well-organised and very useful; the index is thorough and the notes on the contributors are of interest. However, there are two major disappointments in the book. The first lies with the inclusion of the chapter by Caroline Rude, which has little to do with the overarching theme of the book and which, it seems to me, could have done with a little editing. Perhaps it was included because Caroline Rude is, after all, seen as a guru of the profession. The second suspect aspect of the book lies in the way in which the diagram in the chapter by Jean Hollis Weber has been

presented. Whether the faultiness of the diagram originated with Weber in the first place or not, effective copy-editing and subsequent proofreading of the book should have picked up the discrepancies between text and diagram before publication. Still, a great book to have on the shelves, the references alone being enough to ensure this.

Diane Josey

Diane Josey works as a freelance writer, proofreader and indexer for Australian Professional Writers, after a career in education.

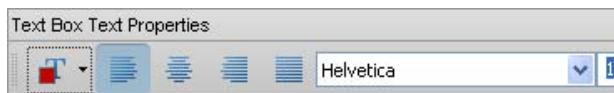
Tips and Tricks

Changing text colour in Acrobat comments

You can change the colour of the text that appears in an Acrobat sticky note, text box and call-out. The Acrobat online help is not much help here. It explains how to change the colour of a sticky note, text box and call-out, but not the text inside them. Here is how it is done:

1. Select the text.
2. Press **CTRL + E** (or right-click an empty area of a toolbar and select **Properties**).

The **Text Box Text Properties** (or **Pop-Up Text Properties**) toolbar appears (shown in part below).



3. Click the first option on the toolbar and choose a new colour.
4. To make your new choice the default for further instances of similar comment containers, right-click inside the container—the sticky note, text box or call-out—and select **Make Current Properties Default**.

Micro-positioning objects in FrameMaker

You can move just about any text object in an Adobe FrameMaker document by selecting the object and pressing **ALT** + an arrow key.

If, for example, you were too pressed, or too lazy, to create a subscript character style, you could convert CO₂ to CO₂ by selecting the 2, reducing its size to, say, 7 point, and then holding down the **ALT** key while pressing the downwards arrow key.

Changing the name of the *Home* breadcrumb in AIR Help

If you are using RoboHelp (versions 7 or 8) to generate online help, you may be experimenting with the Adobe Help Viewer and packaging your project as an AIR file (that is, an *Adobe Integrated Runtime* file). AIR files bring some Web 2 functionality to online help (such as user commenting). AIR is one of the options you have when you generate (or publish) a RoboHelp project (along with Web Help, Flash Help, Oracle Help, JavaHelp and so on).

But whereas with, say, Web Help you are provided with an opportunity in the interface to specify a name for the starting breadcrumb—which otherwise defaults to **Home**—you are not when you go to generate AIR help. Hence, if the topic you want to appear first in your finished help system is called **Home**, you will end up with an initial breadcrumb that reads **Home > Home**.

But there is way to specify the starting breadcrumb:

1. Open the folder that contains your project files.
2. Open `Adobe AIR.ssl` in a suitable text editor (such as Notepad or WordPad).
3. Search for the element named `SSBreadCrumbHomePage`.
4. Change its value to whatever you want the starting breadcrumb to be (perhaps the name of the product you are documenting).
5. Save the file.

The next time you generate AIR help from that project, the starting breadcrumb will be the text you specified in the `.ssl` file, such as **Volvo Maintenance Manuals > Home**.

Miscellany

Vale Swinburne courses

Alas, the technical communication courses at Swinburne University—the Graduate Certificate and the Graduate Diploma—have folded. Current students will be able to finish their studies, but 2011 heralds the end. This should strike all in our profession as regrettable. The sorry state of so much end-user documentation these days is due in large part to the insidious devaluing of our contribution by commerce and industry. The gains we made twenty to thirty years ago are being eroded, and commercial behemoths like Microsoft and Adobe must take some of the blame. Their pitiful documentation offerings have lowered the standard. If Microsoft can make billions a year with so little useful documentation, why should lesser companies bother. Universities still garner some respect, and so should provide a bulwark against the devaluation of our profession. To lose that avenue of professional respect is a worry, as no other university in Australia offers what Swinburne did.

One can't help thinking, though, that some responsibility for the demise of these courses lies with the marketing folk at Swinburne. They published testimonials on the course website that were misleading and barely literate, they failed to attract international interest in the courses' online offerings, and they occasionally misled with what was seen by some as false advertising. Sadly, the splendid work of the teaching staff might have been ultimately thwarted by what might be considered inept marketing.

Style manual (aka the AGPS style manual)

Just over twelve months ago, news came out that a new edition of the much loved *Style manual for authors, editors and printers* was just around the corner. This was welcome news, as the previous edition, published in 2002, was, in quite a few of its recommendations, undeniably out of date.

Many of us responded to the call from the Department of Finance and Deregulation—the manual's publisher—for volunteers to attend focus groups. But then came silence. In May this year, the silence was broken when the Department called for help in the form of a request for expressions of interests, under the title "Exploration of Solutions for the Australian Government's Style Manual". The closing date was 29 June, but then silence fell again. Miscellany queried the Department in October 2010 and was told that "given that work on a 7th edition has not commenced, there is little likelihood that a 7th edition would be available within the next 12–18 months". Bugger!

Swamp the Department with passionate concern via Gov.Style@finance.gov.au.

Odd professional idioms

Idioms are, by definition, idiomatic, so there is not much that can be levelled against them. By being idiomatic they are doing their job of communicating. But sometimes professions adopt idioms that sound clumsy, generate clashing connotations or distract by using prepositions in unexpected ways.

Take indexing, for example. An indexer will, by tradition, precede certain cross-references with *See also*. But outside the profession, similar constructions would sound unidiomatic:

You should *see also* the Uffizi when in Florence.

Most of us, if not all, would have said:

You should *also see* the Uffizi when in Florence.

Computer idiom can also sound odd to the ears of outsiders. Many technical writers follow *The Microsoft Manual of Style* in writing:

On the **Tools** menu, click **Address Book**.

On the Tools menu? How odd is that. We would never say:

On Begonia Road, read every water meter.

Rather, we would say *Read every water meter on Begonia Road*. If we are concerned that we write our steps in the same order that the reader will do them, then *Open the Tools menu and click Address Book* is far less distracting than *On the Tools menu, click Address Book*, as is *From the Tools menu, select Address Book*.

Another *prima facie* oddness is our profession's disposition for the preposition *in*:

In the **Format** window, enter the file number.

Microsoft called these things *windows* because they do resemble, in some measure, the windows in our homes and offices. But the only things *in* those windows are likely to be air bubbles. But we can write *on* windows. Wouldn't it be less distracting to novices if we write *On the Format window, enter the file number*.

Some of us also use *in* when referring to menus:

In the **Print** menu, select **PDF**.

Presumably such folk, on arriving at a restaurant, ask the waiter *What's in the menu tonight?* Ugh.

How much easier might it be for readers if those who need to find new terms and expressions chose those that already matched common usage. If we want people to consider these things as *menus* because they are not unlike restaurant menus, then why not use the idiom we use when looking at a restaurant menu.

But it's too late now. These are all idioms.

Mindstretchers

Geoffrey Marnell

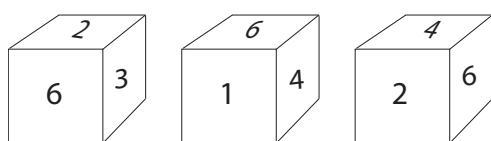
The tortoise and the shadow

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?

Last puzzle

The puzzle in the last issue of *Words* asked:

What is the sum of the digits on the faces of the following cube, shown from three angles?



Solution

The sum of the digits is 22. A cube has six sides and five of the sides—numbered 1, 2, 3, 4 and 6—are already shown. Now, there are inconsistencies in the views given of the cube which suggest that one of these numbers appears twice.

For example, a comparison of view 1 with view 3 will reveal that either the 2 or the 6 is repeated. If you assume the former, you will find that the only side left for the 1 is the side opposite the 6. But this is contradicted by view 2.

Hence the repeated number must be 6, giving an arrangement whereby one 6 is opposite the other, the 3 is opposite the 4 and the 1 is opposite the 2. Adding these six numbers gives 22. QED.

Volume index

Click a volume heading to open that volume.

Volume 2, Issue 1: February 2010

- *Protecting yourself from Microsoft Word*, Geoff Hart, 1
- *Customising FrameMaker dialog boxes*, Dave Reynolds, 5
- *Paragraphing: a vanishing art?*, Geoffrey Marnell, 9
- *Looking at DITA as just another tool*, Suchitra Govindarajan, 15

Volume 2, Issue 2: May 2010

- *Documenting in an agile environment*, Dan Smith, 1
- *Custom keyboard shortcuts in FrameMaker*, Dave Reynolds, 4
- *Introduction to DITA conditional processing*, Dave Gash, 7

- *Language: the core skill in technical writing*, Geoffrey Marnell, 11

Volume 2, Issue 3: August 2010

- *Writing for translation*, Darko Tausan, 1
- *Packaging files with Acrobat*, Geoffrey Marnell, 3
- *Converting legacy MS Word documents*, Bruce Ashley, 5
- *Custom toolbars in FrameMaker*, Dave Reynolds, 8
- *Pitfalls in procedure writing*, Geoffrey Marnell, 12

Volume 2, Issue 4: November 2010

- *Ten technical communication myths*, Geoff Hart, 1
- *Font choice revisited*, Geoffrey Marnell, 9
- *The changing world of product documentation*, Dan Smith, 11
- *Numbering in FrameMaker 9 books*, Mark Ward, 13