



The style files

Write to read breadth first, not depth first

Tony Roberts*

Technical writing is *not* like writing a detective novel: in your introduction, as well as describing the background, tell your reader upfront in plain language your conclusions.

We discuss how to structure the information over a technical document. The fallacy to dispense with is that humans relate to a logical progression: this is false. A progressive logical development, putting in place ‘brick by brick’ your arguments, to a triumphant finale is wasted on most readers. Instead, recognise that readers have varying levels of interest and technical knowledge, thus we should present information in a sequence of increasing technical difficulty. Humans are more likely to keep reading when they feel they are learning something useful, thus present useful information early. To effectively communicate you must write to be read ‘breadth first’, not read ‘depth first’.

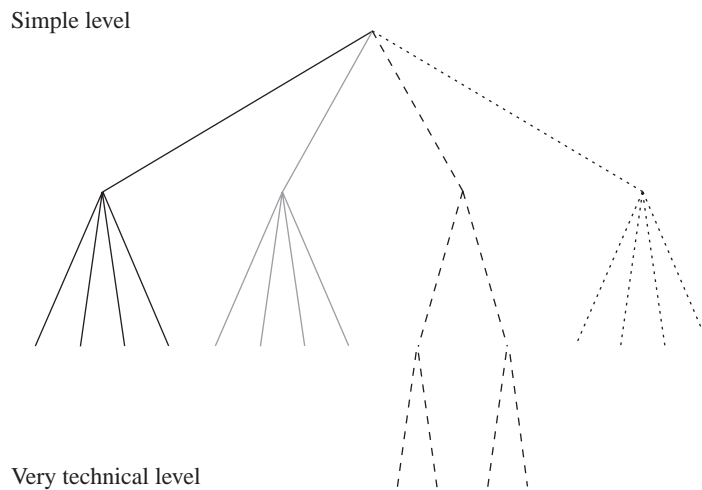


Figure 1. Simplistic schematic diagram of the tree of knowledge in your document. Each major topic you want to discuss is represented by a differently formatted branch, for example: black may be the background; grey, the development of an algorithm; dashed, the performance analysis; and dotted, some examples. The statements you may make will be at different levels of technicality using different amounts of jargon: put the simple easily understood statements higher in the tree; and the very technical statements redolent with jargon deep in the tree.

Consider the knowledge you wish to communicate in the schematic tree structure of Figure 1: technical statements are deep down in the tree; simple statements you

*Department of Mathematics and Computing, University of Southern Queensland, Toowoomba, QLD 4350. E-mail: aroberts@usq.edu.au

may make are higher in the tree. Place a sort of ordering on the tree by having prerequisite knowledge to the left. This is a simplistic model of the knowledge you wish to write about, but I believe adequate in many cases. Now, many people write in a depth first fashion: they set themselves the task to write about ‘blah’, so they write everything about ‘blah’; then move on to the next topic and write everything about that; and so on for all topics in the document. Figure 1 represents each topic by a different styled branch. What does this style of writing mean for the reader? The reader:

- is rapidly enmeshed in deep technicalities about ‘blah’ as they are swiftly led deep into some very technical level material;
- has little idea where the discourse is going;
- gets nothing out of the document because of the difficulty discerning the main points among the multitude of details; and
- quits.

Instead, write breadth first. Write your information so it is read in the order of level of technicality as shown in Figure 2. First, in the title and abstract, a reader sees an overview of the entire document in plain language and including the results. Second, in the Introduction, overview the entire document again, but at a little more technical level. Third, the body of the document records the gory details. Conventionally we also place a conclusion at the end — such a conclusion is out of place in this structure, but nonetheless readers do like a recapitulation (recall that humans are not logical). And lastly, as indicated in Figure 2, appendices include any deviously technical material.

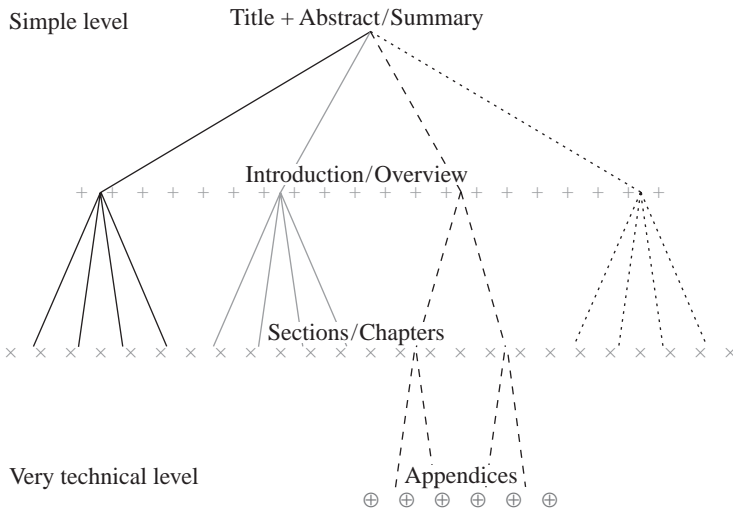


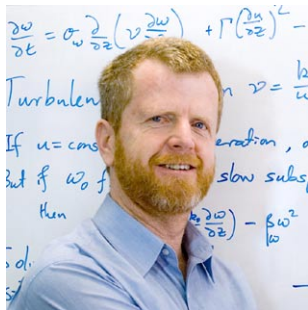
Figure 2. Allocate various parts of a document to the ‘tree of knowledge’ of your document so that the simplest set of statements overviewing your material is read first.

Urging you to ‘write breadth first’ is a misnomer; instead write so the document is *read* breadth first. Usually you will write the detailed technical parts, then winnow out of these the lower-level, jargon-free, introductory and conclusion statements to form the overview in the Introduction. Similarly the title and abstract are extracted as the plainest statements about the setting, your achievements, and how a reader can use the results. Thus at any stage the reader will have an overview

of what you are doing and why — readers will be empowered to place increasingly technical material upon the framework you have already constructed for them. This is the structure to keep readers interested in pursuing your efforts.

References

- [1] Garrett, A. (2000). *Principles of Science Writing*. Technical report. Scitext, Cambridge. <http://www.scitext.com/writing.php> (accessed 3 February 2008).
- [2] Higham, N.J. (1998). *Handbook of Writing for the Mathematical Sciences*, 2nd edn. SIAM.
- [3] McIntyre, M.E. (2005). *Lucidity Principles in Brief*. Technical report. <http://www.atm.damp.cam.ac.uk/people/mem/lucidity-in-brief/> (accessed 3 February 2008).
- [4] Priestly, W. (1991). Instructional typographies using desktop publishing techniques to produce effective learning and training materials. *Australian Journal of Educational Technology* **7**, 153–163. <http://www.ascilite.org.au/ajet/ajet7/priestly.html> (accessed 3 February 2008).
- [5] Strunk, W., Jr (1918). *The Elements of Style*. W.P. Humphrey. <http://www.bartleby.com/141> (accessed 3 February 2008).
- [6] Wheildon, C. and Heard, G. (2005). *Type & Layout: Are You Communicating or Just Making Pretty Shapes*, 2nd edn. Worsley Press.
- [7] Zobel, J. (2004). *Writing for Computer Science*, 2nd edn. Springer, London.



Tony Roberts is the world leader in using and further developing a branch of modern dynamical systems theory, in conjunction with new computer algebra algorithms, to derive mathematical models of complex systems. After a couple of decades of writing poorly, both Higham's sensible book on writing and Roberts' role as electronic editor for the Australian Mathematical Society impelled him to not only incorporate writing skills into both undergraduate and postgraduate programs, but to encourage colleagues to use simple rules to improve their own writing.