Technical Notes, general Series

Technical Writing made easier

rev. 1.1, March 2002

by **Bernhard Spuida, bernhard@icsharpcode.net**
Senior Word Wrangler

Table of Contents

1. Introduction

Technical writing requires clarity of expression and therefore simplicity of language. Technical writing is intent on expressing certain key concepts so that these may be understood as easily as possibly by the intended readers — be they programmers or users. Writing in a clear, concise manner makes not only understanding the text easier for the reader, it also makes your life as a writer of technical documentation easier — especially when you are not a native speaker of English.

When talking about algorithms, or sequences of events in a program, absolute clarity of writing is not only needed in the code discussed; but also in documenting this particular program for our fellow programmers and users. We need to attain the same level of clarity of expression in both cases, otherwise readers will to turn to other programs, which are more accessible on the level of understanding and therefore easier to use or extend.

In this short guide, we will cover some of the basic concepts that lead to good (technical) writing. You will certainly discover more such rules and concepts as you practice the writing skills gained out of this set of notes. And also, read! Read a lot, and read varied writing, conscious of the ways language is used in the texts you read. Have your own writing read and criticised by friends and fellow professionals. Pay attention to these criticisms.

You will see that our colleagues, just like the computers we program, require a specific syntax to be adhered to if we want our instructions to be understood. And as in programs, human language text may be straightforward or convoluted, leading as in programs to variations in performance. So here we go.

As this is intended to be a 'work in progress', additions will be made whenever necessary. I am also always happy to receive suggestions and feedback.

2. Theory

The understanding of written text depends on three distinct components:

- Legibility
- Readability
- Comprehensibility

The first of these components is of no concern to us, as it is a responsibility of the layouters and typesetters putting our writing into its final form.
The second of these we will deal with, as it is vital to having the reader actually *read* our document, hopefully in full.
And lastly, the third component is essential to ensure that our reader will *understand* the purpose of our writing.
These two components will be discussed in separate sections, even though some of the issues raised may be pertinent to both.
In addition, we will also look at issues of style — some of writing's do's and don'ts , as even the prose of technical writing does not have to be equivalent to a blunt axe when it might be an instrument of precision.

Should you, kind reader, have suggestions for improvement to these pages, please let me know: bernhard@icsharpcode.net

3. Readability

The concepts of readability and comprehensibility imply that the act of reading beyond the physical act of seeing and deciphering characters and chunks of text is vastly more complex.

As the next step beyond this 'raw' level of input, we need to assume a process of tokenising, akin to what a compiler does with the source code of a given program.

This process of tokenising is what readability is concerned with. Thus, our writing will need to meet a number of requirements to successfully pass this stage:

1. The sentences must be well formed syntactically
2. The sentences must not exceed a certain length
3. The sentences should not be below a minimum length
4. Recursion must be kept to a minimum
5. The choice of words should vary

If a technical text is unreadable in the reader's eye, he will quite probably assume that the product described in this text also is of inferior quality. Code is a language, just as the language of the documentation is. Not writing well in documentation implies faults in coding style.

Therefore, readability is an absolute requirement for documentation of successful products.

3.1 Well formed Sentences

By well formed sentences, we do not merely mean that the sentences should conform to grammatical rules of the English language, but also that they are clearly built. We will now look at some negatives and discuss solutions:

This sentence no verb

Glaring grammatical errors such as omitting a vital component of the sentence — in this case the verb — should be avoided at all cost. Read out loud, whenever in doubt. Usually, these mistakes occur in longer, more convoluted sentences. Check these twice when they cannot be rewritten in split-up form.

This sentence does a verb have

Never, ever try to transpose a grammatical construct of your mother tongue into a literal English equivalent — even more so in cases of colloquialisms, as above! If you are able to translate a sentence word by word back into your mother tongue, you most probably made a severe mistake or two in writing it. Read texts by native speakers of English. Rewrite your own text next, and then reread it.

In this case, we see that there is, as such, a larger than necessary number of commas.

Punctuation should be kept to a minimum. It is not necessary to put a comma wherever it looks right. They often are not. Especially clauses of the 'so that' type can do perfectly well without commas. This rule of course also holds for all other punctuation. And never, ever, try to transport punctuation rules from your native tongue to the English you are writing.

Of course, many more cases of sentences not well formed might be constructed, but quite probably you will find enough of these when looking through various pieces of writing. And not only will you find these in non-native writers of English. Therefore, again — read what others wrote!

Of course, most of the further examples of section 3 also are malformed in our wider sense.

3.2 Overlong Sentences

Often we encounter sentences which run on too long. Understanding such sentences is extremely difficult, as short term memory has a very limited capacity. Similar to the rule that telephone numbers may not have more than 5 digits plus/minus two, sentences should not exceed a certain length.

It is given as a rule, which however is not the only such rule you may encounter, that sentences should not exceed a desirable length of ten to fifteen words, never should fall below seven words or extend beyond the ultimate limit of tolerable length reached at twenty words, even though longer sentences may be found in high literature, where even punctuation as it is used in this example to facilitate reading is oft omitted in novel experimental ways.

This of course is an example that runs somewhat longer than what you would expect to find in your own writing. But read your own texts again and you will quite possibly find one or two of these abominations, describing say, a complex chain of events and their handling. A complex train of thought can only benefit from being broken down into sentences of convenient length. Temptation to ramble on in one long sentence may be great. Resist. Your logic will benefit. Also cut out anything not necessary to the immediate cause at hand. To quote Strunck and White's third rule:

Omit needless words.

3.3 Short Sentences

Short sentences are easily read, but tend to look breathless and overly excited.

Sentences may be short. Then they are easy to read. And understand, too. But they look cheap. And breathless. As well as leaving the reader restless.

Not much needs to be said here, as these above sentences illustrate the point to be made. If a thing is worth saying, it is worth saying it well, not chopping language to pieces. Human language is not a RISC language.

In general, try to vary the length of sentences in the limits given in the negative of subsection 3.2 above. Interesting writing depends on well dosed variations of length and choice of words — for examples of the latter, see below.

3.4 Recursion

Sentences often turn into a quasi-circular case of recursion while reading them when the references made in the sentence to the respective objects and subjects are left unclear by using the same pronoun to describe these subjects and objects.

It is not easy to understand it when it is unclear what is referenced by 'it' – it by now should be clear what it is supposed to mean, isn't it?

In such a case, replace one — preferably the first — instance of each subject/object referenced by 'it' with its actual name. This will make reading much easier. Recursion of this type often also is an indication of laziness on the writer's part, as it hints at an unwillingness to formulate a thought properly. This however is only a short term saving of effort, as readers probably will contact the author asking for clarification and thus causing more unproductive work on the writer's part than necessary.

Another form of recursion will take place in the reader's mind when he is confronted with convoluted sentences containing insertions, ellipses and other rhetoric figures. All of these will need some place on his 'stack' — and a reader's stack is shallow. 'Pushing' and 'popping' more than three stack levels usually ends in disaster … for the message of your sentence!

The reader, that is, the intended recipient of our text, a hopefully clearly written and logically structured document, will, if he is able to fully understand our prose, without difficulty come to a safe assumption of what any given sentence, such as this a one, conveys, to him, the reader, by the way of meaning.

The above sentence is grammatically correct, but will most probably provoke a 'stack overflow' in our average reader. Such convoluted[1] writing should be avoided wherever possible. Just as recursive code, text may be 'flattened out'. Take a monster such as the one above apart. Shorter sentences make easier reading. If this is not possible, try to keep related parts of the sentences as close to each other as possible.

3.5 Choice of Words

Reading is a process that takes its toll on the reader. This task should therefore be made as easy as possible for him. Making reading an easier task, if not a pleasure, can be achieved by varying the vocabulary used to describe the topics at hand.

Using the same words all over to describe the same things again and again is not pleasant even more so when we can use different words to replace those same words we are using again and again to describe the same thing in the same words.

For any given word, at least one synonym will be available. Do not hesitate to use a thesaurus. Also, do not use the exact same phrasing again and again and again, unless it is intended to convey some artistic intention — this however is almost never the case in technical writing. And:

Never use the same opening words in two or more subsequent sentences. Repetitive writing is the enemy of all reader's interest.

4. Comprehensibility

In the complex process of reading, the step following the 'tokenisation' of the text is the actual 'parsing' — understanding what these symbols and their relations mean. A clear separation of these two steps however can not be made.
A great portion of comprehensibility issues already was covered when we discussed recursion

[1] Go ahead, look up 'convoluted' in a dictionary. Now. Do this as well for any other word you may not know in this or any other text.

above.

Recursion is the enemy of understanding

An understandable technical document always follows a logical structure. Any topic discussed is based on the preceding topics. If a new concept is needed for the topic at hand, it needs to be introduced before using it in dealing with this new topic. This holds true for any level of detail of the document at hand, down to individual sentences.
The basic steps are:

1. Definition
2. Assumption/Theorem
3. Explanation/Proof
4. Conclusion

Of course, the classic structure of '*thesis, antithesis, synthesis*' may be more appropriate for certain topics, such as discussion of architectural decisions, but generally the above sequence is exactly what we need.
On the 'atomic' level of a sentence, its logical structure is governed by raw grammar. Therefore, a good working knowledge of grammar is absolutely necessary for getting our ideas across to the reader as we mean them to be.

4.1 Definition

In this segment of the document, all terms and concepts necessary for the following should be defined. In some cases, reference to previous material in the document is sufficient. Referring to later sections is to be avoided at all cost. If it should for some reason be necessary, the definitions may be placed in endnotes or a glossary at the end of the document. This should be clearly indicated at the very beginning of the document. Footnotes are not intended for the purpose of definitions. They are a place for further explanations or material of a 'non sequitur' nature[2].

4.2 Assumption/Theorem

The task of this segment is the presentation of the idea or concept this particular document or part of a document is supposed to deal with.

Make a simple and clear cut statement of where your argument starts and what will be the intended outcome. No why or how should be given here. The why should be clear from earlier portions of the document, the how is the subject matter of the next section.

In a user's guide, this is where an indication of the function to be explained should be given.

4.3 Explanation/Proof

This segment of our document deals with giving justification for the idea put forth in the previous section. It may be of purely argumentative[3] nature — say, defending architectural decisions — or come close to mathematical proof in style. In the case of a program implemented practically, this is the place for explaining the workings of it step by step. Or in the case of a user's guide, to explain the interface and sequence of steps necessary for completing a given task.

---

[2] Latin: 'is not followed' i.e. something that is not important for the understanding of the subsequent text, but merely a side track of additional information.

[3] Just read some classic rhetoric texts to understand what I mean. Socrates should ring a bell even without reading…

4.2 Conclusion

For the document to be successful, a conclusion must be given, reiterating the above steps in a shortened form. This will reinforce the impact of the material presented on the reader. The human mind, unlike a computer, needs to be told several times before committing to a certain course of action.

Repeat the central message of what you write several times. The human mind is not at ease when confronted with a ‘fire and forget’ type of message.

Now, without looking back at what was written above give a summary of what was said in section 4. It won't be repeated here. Can you recall the topic of the last subsection of section 4? What number was that subsection?

See?

5. Matters of Style

This section will be concerned with the little things that will hopefully turn acceptable technical writing into good writing. There is a number of seemingly small and unobtrusive 'do's' and 'don'ts' that need to be watched out for consciously even though they seem to be obvious.

The following material is *not sorted by relevance*. You will just find things as they came to mind, read through it and consider it as a wish list for good writing.

5.1 Title

The first thing our reader sees is the title. Therefore, the appearance of the title is of great importance for the impression our text leaves. The dominating factor is capitalisation of the words in the title. There are several 'schools of tought' as far as this is concerned:

1. **All Words In Titles Are Capitalised**
2. **Only the first word is capitalised**
3. **nothing is capitalised**
4. **First Word and all Nouns are capitalised**

Of these, I personally recommend number 4[4], as it is the style usually choosen for scientific publications. Numbers 1 and 3 are usually bad for the readability of the title. In any, case, once you have chosen a style, stick with it. **No exceptions to the rule!**

When numbering titles, consistency is also a must. If you settled for a certain scheme, stick with it! Usually the numbering styles suggested by your text processing software – be that LaTeX, Open Office or what ever else – of choice represent sensible schemes.

5.2 Big Words

Don't go above your station. It may well be tempting to employ vocabulary gleaned from some obscure manual of language, obfuscating your intent by billowing clouds of rhetoric smoke, this however will most certainly induct the reader's total non-comprehension of your text.

By now it should be clear what the name of the game is: in the case of two given words of the same meaning, preferably use the simpler one. With one exception: should the 'bigger' word be clearer as to its meaning, use that.

And when you find yourself in the 'Big Word Game', also check the length of your sentences. There seems to be a correlation between the two.

However, you will find that using only the smallest possible word to refer to a concept will making your writing look dull. A bit of variety never hurts.

[4] And use it for this Tech Note.

5.3 It's...

It's often a matter of confusion for non-native speakers to decide when to use 'its' and when to use 'it's'. The former is the genitive of 'it', the latter is a contraction of 'it is'. Simple. When in doubt, check back here.

And preferably write 'it is', as this makes for better style than 'it's', without too much extra typing effort. This way, the problem of what form to use will vanish automagically, anyway.

5.4 An 'a'

It is of course somewhat tempting when trying to write good English to use 'a' and 'an' in accordance with the simple assumption *that 'a' is to be used before consonants and 'an' before vowels*. This is not in accordance with the rules of proper English. Simple (assumed) rules usually have exceptions to prove them in general. The exception in this case is with vowels:

- *An apple*
- *An exception*
- *An idea*
- *A useless rule* but
- *An uppercase letter*

This is of course a mere rule of thumb[5]. For the full story refer to a style manual such as 'The King's English'.

5.5 Do not use 'don't'

The use of contracted negated verbs is something that should not be done in good technical writing. Writing out a negation in full does not take much effort on the writer's part and leaves the impression of a writer caring about his style. Should he also be the coder of the program documented, attention to such details will also affect the perception of the quality of his coding style by the reader.

So: **no** won't, can't, don't, isn't, aren't, couldn't, shouldn't

And **never, ever**: ain't, shan't

5.6 Can, could, etc.

Coming from a German language background will lead to a very specific problem[6] with the various meanings of the versatile German auxiliary verb 'können' when translating it into English.

- *Can*: the ability to perform an action — *after having initialised foo, we now can call bar(baz)*
- *Could*: a possibility, or rarely in the case of technical writing, the past tense of can — *in the case of an exception being thrown, we could catch it, or dismiss it.*
- *May*: a speculative possibility — *in future versions of foo, we may implement bar().*
- *Might*: an alternative — *instead of calling foo(), we might call bar() when the following conditions ... are fulfilled.*

[5] I guess you see where the general drift goes?

[6] Problems specific to other languages might be discussed in future versions of this document.

- *Should*: a future, not immediate, option — *if ever foo becomes obsolete, we should consider implementing bar*

5.7 Nativisms

In general, be suspicious of 'too obviously natural' translations of words from your native tongue into English. E.g., the German 'Konkurrenz' will translate to English as 'competition', not as 'concurrence' as might be naively assumed. Get a good dictionary and check it often, read English texts you are already familiar with in your tongue — no matter whether that text is originally English translated to your native tongue or vice versa[7]. Preferably start with some favourite book of yours — be that 'The Lord of the Rings', 'Grimm's Fairy Tales' or 'Gödel, Escher, Bach' or even comic strips or whatever else has taken your fancy.

Understanding words properly matters. When in doubt about the actual use of a word with several potential meanings in translation, check with a native speaker or refer to a good reference dictionary such as the 'Oxford'. Some online resources for dictionaries are given in section 7.

5.8 Ego Trip

Never, ever, use the word 'I' in technical writing. It is anathema. We are dealing with objective matters, not personal opinions. 'I' may only be used when voicing very personal opinions which usually do not belong into a document of technical importance – when discussing design decisions made by you, 'I' may be used legitimately, but not when describing a technical detail such an in interface. When engaged in a flame war, 'I' is perfectly fine[8], but not anywhere else near a technical topic.

Should you however feel a burning urge to use 'I', switch over to the passive voice or third person writing.

Turning to impersonal writing worked well for Caesar, when he was giving *his* account of *his personal* accomplishments in the Gallic wars — this account is considered classic literature, even though it in fact is very technical writing. The passive voice is to be avoided in general, as it makes for tiring reading, but in such cases as we are now discussing, it is preferable to endless 'I think…, I consider this to …, I did…'.

Whenever possible, use 'we' to forge a bond between you and the reader. This will give a feeling of 'being in it together'. Classical technical writing is very much bound and impersonal style in general. This nowadays is no longer true. But still, technical writing is not a 'family thing'!

5.9 When to use 'if'

Conditional clauses may be begun with 'when' or 'if'. Confusion sometimes comes from the problem of when to use which. With a bit of thinking, the distinction between these two cases is simple:

- *When*: used in cases of a temporal conditional — *when the sun rises, the cock crows*
- *If:* used in the case of a causal conditional — *if we manage to get out of this text alive…*

5.10 This Sentence does overdo it

Using 'do' as an auxiliary verb usually does more harm than help. It tends to overemphasise

[7] The former case is preferable, of course.

[8] As for flaming, IMHO is a pleonasm, IMO suffices perfectly well — **no personal opinion is humble**. This now was a perfect example of a 'non sequitur' footnote.

sentences — especially with 'weak' verbs such as 'have' it gives the impression *of speaking just that little bit too loud*. Using 'do' as in the introductory sentence of this paragraph as a verb by itself is in order. But it does give a certain obnoxious quality of overassertiveness to your writing if you do use it as an auxiliary verb when you definitely do not need to use it to say what you do want to say. Now that does settle this point, does it not?

5.11 Time is on our side

Always try to keep the (grammatical) tense of technical documents in the present. Past tense ought to be out of the question, as we generally are not concerned with past developments. The future usually also is beyond our scope, as we do not yet know what future revisions of our program will bring.

5.12 Consistency

Be consistent in the use of either British or US-english spelling. Never switch inside any given document. Examples of difference in spelling between those two are:

British	*US*
Colour	Color
Co-operation	Cooperation
Customisation	Customization

Also, once you choose a given technical term to mean one thing, use it only in that one sense. **Do not redefine!**

5.13 Editor's pet peeves

At some point in any writer's life comes the moment where his work is submitted to an editor. This breed of person does have a lot of experience with language and the common weaknesses of the technical writer. Over time it has become clear that the same mistakes are made again and again. So, save them time and yourself humiliation by following the list below. It contains additional material not covered above.

To make things easier, the list is organized by topic. The most important issues or the most commonly misused terms are **bold**ed. The words in this list are supposed to be spelled as printed here, including capitalisation, hyphenation or lack thereof, and separated or not as given. Also note that US-english is the reference here, as opposed to the other text[9], as almost all technical writing is published by American publishing houses.

5.13.1 Grammar and Logic

- allow vs enable: People allow; objects enable. Some editors are more stringent about this rule than others. - Georgie *allowed* me to borrow his CD. The program *enabled* me to create a stunning document. You might also consider to use *let* as synonym for *allow*. - He *let* me borrow his CD.
- although vs while vs whereas: Use *while* only when describing the action of doing two things simultaneously. Use *although* and *whereas* for contrast.

[9] Now, that is my personal decision I choose to make as the author of this Tech Note.

- **Figures**: The first reference to a figure should always be before it is presented. Ensure that the spellingof terminology in figures is same as in the text. Captions (please remember to include them) should be informative sentences. - bad caption: The File Open dialog box. good caption: In the File Open dialog box we ... When referenced in the text with a specific number, the term 'Figure' is always capitalized.
- graphic (n): The noun form usually takes a singular verb.
- graphics (adj): Use *graphics* as an adjective in relation to the field of graphic art or graphic design - graphics file is ok.
- **once** vs **after** vs **when**: Use *once* for time references. Use *after* to show that an event has already taken place. Use *when* to show actions or events that are occurring simultaneously.
- over vs more than: *Over* denotes crossing a barrier or exceeding a limit. *More than* qualifies a number. - She has worked more than 60 hours this week.
- preceding vs previous: *Preceding* means immediately before in time and place. *Previous* means simply earlier, prior, or before, but not necessarily immediately before. For example, if the current month is June, the preceding month is May; previous months are January through April.
- since vs because: Use *since* for time references. Use *because* when explaining the reasons why something happened.
- **There is, There are/ Here is, Here are**: Sentences starting with this construction are usually more clear and less wordy if they're rewritten.

5.13.2 Spelling and Terminology

- **above, below**: Avoid this terminology when referring to figures or other references. Instead, be specific (see Figure 24.2) or use *preceding*, *following*, *next* and so on. See also *preceding*.
- backward, toward, forward (no "s" on any of these words)
- check box, check mark
- back up (v); backup (n)
- clip art
- Clipboard
- CompuServe
- Control Panel
- Ctrl key (spelled as it is on the keyboard)
- Ctrl+Alt+Delete vs Ctrl-Alt-Delete: On the PC, keystroke sequences are separated by plus signs. In Mac texts, sequences are separated by hyphens. This however, is subject to variation depending on the editor's preferences.
- desktop

- **disc** vs **disk**: Use *disc* when referring to a CD or an optical; Use *disk* when referring to a 3 1/2- or 5 1/4-inch disk. Never use *diskette*.
- double-click, right-click, left-click (always hyphenated as verb)
- drop-down (adj, n); drop down (v)
- **email:** no hyphen, no capitalization
- **filename**
- **home page**
- hotkey (one word): Make sure that hotkeys are marked in the chapters. Hotkeys are indicated typographically.
- HTML: Hypertext Markup Language (notice that the "t" in *Hypertext* is lowercase)
- inline
- intranet is lowercase; Internet is uppercase
- ISP: Internet service provider (no need to make the "s" and the "p" uppercase)
- **keyword**
- log in/login vs log on/logon: Both forms are acceptable, but be consistent. As a verb, *log in* or *log on* are two words. As an adjective, *login* and *logon* are one word.
- **Measurements**

bps	bit per second
GHz	gigahertz
Hz	hertz
KB	kilobyte
Kb	kilobit
Kbps	kilobits per second
KHz	kilohertz
MB	megabyte
Mb	megabit
Mbps	megabits per second
MHz	megahertz
ms	millisecond

- As for Kilobytes, be consistent in using either decimal (1K=1000) or power of two (1K=1024) notation. The former is to be preferred for physics, the later for computer science contexts. This also holds true for higher powers (mega, giga, tera, peta).
- menu bar
- **multi, non, re, sub, and co** prefixes: It's almost always safe to assume that you

don't need to hyphenate these words. (For example, the correct spelling is multicolumn, not multi-column.) A few exceptions to note are as follows:

resort vs re-sort: *Resort* means a place of retreat; *re-sort* means to sort again.

recreate vs re-create: *Recreate* means to cut loose; to play. *Re-create* means to create again.

resent vs re-sent: *Resent* means to show displeasure or indignation because of a feeling of being injured or offended. *Re-sent* means you sent something again.

There are a few exceptions in British usage though, such as co-operation. Settle the reference spelling context once and for all when you begin writing.

- **Net** - uppercase when referring to Internet, **.NET** or **.net** - all upper or lower case when referring to the Microsoft technologies, be consistent with the choice for the latter term.
- newsgroup
- **offline, online**
- **offscreen, onscreen**
- pathname
- Plug and Play
- PostScript
- pull-down menu
- ScanDisk
- **scrollbar**
- set up (v); setup (n, adj)
- **Spacebar**
- status bar
- Tables: Be specific when titling tables, and use sentence style capitalization (only first word and any proper nouns are capitalized) for table heads. As with figures, the word *Table* should be capitalized when a specific table is being mentioned (Table 4.5), but lowercase when the reference is generic (The following table...). An ending period is no longer used after the table number in any series (Table 4.5 is correct; Table 4.5. is incorrect).
- **taskbar**
- **terminology**: Consistent use and capitalization of terminology is important. A novice reader as well as a general tech editor won't always have the knowledge or the software about which you write and thus won't always be able to catch technical inconsistencies.
- title bar
- ToolTip, ScreenTip
- TrueType font

- UNIX
- up-to-date (always hyphenated)
- **URL**: uniform resource locator ("u" stands for uniform, not universal).
- Usenet
- **username**
- **Web** - is always uppercase)
- wizard: Use uppercase "w" when talking about specific wizard and lowercase "w" when talking about generic
- worldwide (one word, unless referring to World Wide Web)

6 Recommended Reading

Of course no complete list of books on style can be given here. The selection must by necessity be a personal one. So I will just list a few books I found useful for myself:

Strunck & White: 'The Elements of Style'
'The Chicago Style manual'
'The King's English'
Stephen King: 'On Writing' (Good ideas about writing in general, not technical. And a nice bit of autobiography, too.)

Some books that are well written and will give some idea of what can be done:

Robert M. Pirsig, 'Zen and the Art of Motorcycle Maintenance' (An interesting book about motorcycling, ,the human psyche, 'quality' and — a little — about technical writing, all well written)
Douglas R. Hofstadter: 'Gödel, Escher Bach – an Eternal Golden Braid' (Some of the finest writing ever on the concept of 'Strange Loops' which manages to get some rather technical points of AI and programming across without being boring)
Donald Knuth: 'The T X book' (One of the finest software manuals ever written. Even the more
e
arcane aspects of TeX are clearly explained in an entertaining way)
The VAX manuals, a.k.a. 'The Orange Wall' (Tons of paper, the best set of software documentation ever. Much better than most man pages and galaxies ahead of PC-software documentation)

7 Online Resources

Some online translation and dictionary resources:

http://www.tecnologix.net/laixicon/
http://dict.leo.org/
http://www-tgw.dfki.de/~winter/lang/anglizisms-de.html
http://www.linguadict.de/
http://www.dictionary.com/
http://wombat.doc.ic.ac.uk/foldoc/
http://www.onelook.com/browse.shtml
http://www.tuxedo.org/~esr/jargon/

As you will notice, some of these refer to issues specific to the German language. Even if you are not a native speaker of German, these will illustrate some of the basic problems with nativisms.