Tips for Good Technical Writing

Tom Hastings

File: good-writing-tips-v01-030127.doc
January 27, 2003

This document contains tips for good technical writing. Following these tips will help improve the understandability of your specifications and so improve conformance and interoperability.

Table of Contents

1. Test each sentence for understandability

Test each sentence. Be your own Devil’s Advocate against each sentence. If there are no ways or more than one way to understand the sentence and its implication, re-work the sentence until there is only one way to understand it. Follow the “SMART” sentence testing criteria:

1. **S**pecific - Is the sentence clear, unambiguous, and open to only one interpretation?
2. **M**easurable - Is there a quantitative way of determining if the requirement is satisfied? Could be through an actual measurement, an enumeration, inspection, or any number of other means.
3. value **A**dding - Is this requirement something that adds value? It could be specific, measurable, real and testable, and not make any positive difference to the actual Interface.
4. **R**eal - Is this a real requirement, or just someone's opinion or pet issue? We want to make sure Interface requirements are validated against some customer need.
5. **T**estable - If you were to test this, how would you do it? If you know in advance how you'll verify, you got the Interface requirement right.

2. Define your terms carefully

Define the terms that you are using so that the reader is clear what you mean. Don’t assume that your reader has the same understanding of terms that you do. Include the terminology section up front, so that the reader is clear about your terms. In order to differentiate your terms from the

generic meanings for the same word, it is important to Capitalize the first letter of each word in your terms in order clearly indicate that you mean the word in the sense you want. When checking your sentences for ambiguity (following writing tip #1), try to think of all possible meanings to the words that your readers might have. If there are multiple meanings, either clarify the sentence or define a specific term and use it consistently.

Your terms SHOULD be the ones the are commonly used in the industry, rather than being different words. Some authors misguidedly use different words in order to avoid confusion with the common words. However, that makes their documents much harder to understand. Instead, use the commonly used words and define them in your terminology section with the specialized meaning that you want. For example, the work “attribute” is commonly used to indicate a named feature of an object with a data type and one or more values. You SHOULD define Attribute in your specification and define exactly what it means in your specification, possibly saying how it differs from that term used in other contexts. If necessary, you can put an adjective as part of your term or in front of the other usage, if you have to use it in your specification. For example, XML uses the term attribute in a specialized way as part of an XML tag with the syntax: attribute-name = value1 value2. If you need to refer to the XML attribute and attributes in your speciation, define XML Attribute and Attribute as two distinct terms in your terminology section, rather than introducing some other term for your Attribute.

3. Avoid the passive voice

Avoid the passive voice. This tip is a special case of tip #1 . That is, instead of writing: “The job is deleted" you should write: “The server deletes the job", which is more specific and absolutely necessary for a specification that talks about both a Producer (client) and a Consumer (server). Otherwise, a reader could misunderstand and think that it was the client that deletes the job. It’s even more critical when the sentence contains a conformance term, such as “The job MUST be deleted”. Who MUST delete the job? The client or the server?

4. Be careful of the word “or” (choice for whom?)

Be careful of the word “or”. This tip is another special case of tip #1. The word “or” means a choice for someone or something. Does it mean a choice for the implementer or for the implementation? Is it a choice for the Consumer or the Producer?

- Consider the sentence: “The Device MUST return a or b.” Who gets to choose? Must the implementer support returning both and the implementation chooses which one based on circumstances, or MAY an implementer choose to always return one fixed value?

- Consider the sentence: “The Device MAY return a or b.” This sentence is even worse, because it could mean that a conforming implementation doesn’t have to return either value.

5. Distinguish between “support” versus “send” for Producer

For Producers, clearly distinguish between “MUST support xxx”, that is the Producer contains code that emits xxx across the Interface to the Consumer on some occasions, and “MUST send xxx”, that is the Producer emits xxx across the Interface to the Producer on *all* occasions.

6. Avoid the use of pronouns

Avoid the use of third person pronouns, such as “he”, “she” “it”, or “they”. Instead, use the noun that is intended. Otherwise, the reader isn’t sure whether the pronoun refers to the subject or object of the previous phrase or sentence. Also it slows down the reader if the reader attempts to

determine which you meant. For example, don't write: "Producers MUST accept all instances. They NEED NOT conform." Does "They" refer to instances or Producers? The second sentence is ambiguous and the reader is slowed down trying to figure out what you meant or worse, misinterprets the sentence. Instead, write: "Producers MUST accept all instances. Instances NEED NOT conform." As another example: "The Producer MUST shut down the Consumer. Then it MUST un-register." Who MUST un-register"? The Producer or the Consumer? Just re-state the antecedent, whichever you intend the sentence to apply to, instead of using "it".

7. Include the "why" justifying "what" statements

Include the "why" justifying "what" statements for all of the following reasons:

1. Including "why" helps the reader to understand the "what".

2. Also the reader is much more likely to remember the "what" and follow the standard, if they understand the "why".

3. Furthermore, writing down an explicit "why" forces you to reveal the reasons for such a decision. If you can't articulate a "why", perhaps you didn't have a good reason for the "what" and there are other alternatives that you should consider.

4. Finally, being explicit about your "why" will allow reviewers to see if they agree with your reasoning about your "what" or whether there are other better "whats".

The "why" information can be included as a phrase ("..., in order to ..."), a parenthetical remark, a "Note: ...", or information in an Informative Appendix. Some standards use the notation: "Rationale: ...", instead of "Note: ..." so that it is clear that the statement is explaining why a decision was made and isn't a conformance requirement. For example, consider the following sentence: "You MUST capitalize the first letter of each word in your terms throughout the document, so that the reader knows that you are referring to the specially defined term." This sentence explains both the "what" (capitalize) and "why" (so that the reader can distinguish the special terms from ordinary use of the words).

The following is a real world example of the mistakes that can be made because a why wasn't explicitly written down so that reviewers could consider alternatives. In IPP/1.0 we decided to make the Print-Job operation REQUIRED for Printers to support and the Create-Job operation OPTIONAL. The unspoken "why" was that Print-Job creates a single document Job and Create-Job, in combination with a separate Send-Document operation for each document, creates a multiple document Job; we all agreed not to REQUIRE Printers to support multi-document jobs. However, the down side to our IPP/1.0 decision is that the client has to send the data as part of the Print-Job operation so that if the Printer rejects the Print-Job operation, the client still winds up wasting time sending the document anyway. In IPP/1.1 we merely allowed a Printer to support Create-Job and Send-Document for single document jobs and provided a means for the client to determine whether or not additional Documents would be accepted with additional Send-Document operations for the Job. Had we written down the "why" for Print-Job being REQUIRED and Create-Job being OPTIONAL, IPP/1.0 reviewers would have pointed out the mistake allowing us to fix the problem in IPP/1.0 instead of having to wait for IPP/1.1. In fact, in order to eliminate one more interoperability problem, it would have been even better if the Create-Job was REQUIRED (as well as Print-Job). Then a client would be free to choose either Print-Job or Create-Job and interoperate with any Printer.

8. Don't specify conformance in a Note or Informative Annexes

Don't use MUST, SHOULD, and MAY conformance terminology in a "Note: ...", "Rationale: ...", or Informative Annexes that is explaining the "why" of conformance statements.

9. Use lots of examples

Use lots of examples. Examples allow your reader to test his understanding of statements. In fact, for many readers, if they cannot construct an example in their minds, they do not understand a general statement. If you give the reader an example, you are helping him along the road to understanding. Also examples help you test that your statements are understandable, rigorous, and correct. If you can't generate an example from your statement or there might be unintended examples generated from your statement, you'll know your statement needs more work.

Here's a real example where the failure to include examples resulted in misunderstanding the descriptions. The Printer MIB [RFC1759] has the following three attributes for Interpreters in the Printer:

prtInterpreterLangLevel: The level of the language which this interpreter is interpreting or emulating. This might contain a value like '5e' for an interpreter which is emulating level 5e of the PCL language. It might contain '2' for an interpreter which is emulating level 2 of the PostScript language. Similarly it might contain '2' for an interpreter which is emulating level 2 of the HPGL language. [Editor's note: according to writing tip #11 below, the Printer MIB definition should have used "that", instead of "which" every where.]

prtInterpreterLangVersion: The date code or version of the language which this interpreter is interpreting or emulating.

prtInterpreterVersion: The date code, version number, or other product specific information tied to this interpreter. This value is associated with the interpreter, rather than with the version of the language which is being interpreted or emulated.

The **prtInterpreterLangLevel** attribute had examples, and so is used by most implementations consistently, but **prtInterpreterLangVersion** and **prtInterpreterVersion** are used very inconsistently by various implementations, if they are used at all, since no examples were included.

10. Avoid repeating the same conformance statements

Avoid repeating the same conformance statements in various parts of the specification, since you may forget to update all occurrences, if you improve or change one occurrence of the statements. Instead, use cross-references generously to avoid repetition. Use MS-WORD cross-references, so that they will always be correct, even after sections are added or removed. If the cross-references are inserts as a hyperlink in MS-WORD then they will also be hot links in the PDF file, if you use Adobe™ PDFMaker (which uses Distiller) rather than using PDFWriter. With hot links, the reader (whether reading the .doc or the .pdf file) will be able to jump to your cross-references.

11. Use "which" versus "that" properly

Use "which" versus "that" properly. A "which" phrase is a parenthetical clause that is giving additional information but is *not* restricting the class of items, while "that" is restricting the class of items. You can easily test whether a phrase is restrictive or non-restrictive: If removing the phrase does not affect conformance, then the phrase is a which phrase and SHOULD be set off with commas to indicate that it is a parenthetical remark. On the other hand, if removing the phrase would affect conformance, then the phrase is a that phrase and MUST NOT be set off by commas, since it is not parenthetical. For example, "A Consumer MUST reject a JDF that is non-conforming" is a proper use of "that", since not all JDFs are non-conforming. An another example, "It is important to conform to JDF, which is a new standard" is a proper use of "which", since the which clause isn't restricting the class of JDF, but refers to all instances of JDF.

12. Avoid the vague word “use” or superfluous phrase "for use"

Avoid the vague word “use”. Use an action verb, especially if it implies a direction across the Interface you are specifying, such as “send” or “include” for a Producer and “support” or “accept” for a Consumer. Bad example: The client MUST use the user’s name in the request. Better example: The client MUST supply the user’s name in the request.

Often the phrase "for use in" or "for use with" is superfluous. Be more specific and remove the unnecessary phrase. Bad example: This attribute is intended for use in identifying document formats that are not portable. Better example: “This attribute is intended to identify document formats that are not portable.” Or even more simply: “This attribute identifies document formats that are not portable.”

13. Avoid vague sentences

Avoid vague sentences. For example, instead of the following vague sentence in ISO DPA 10175: “If no distinction is specified, it is treated as both logical-and-physical.” be much more specific: “If the printer-realization attribute of a print object is missing, the server that manages the print object MUST treat the print object as if the printer-realization attribute has the value "logical-and-physical.”

14. Additional general good writing tips

1. Avoid long sentences. When in doubt, break it up.

2. Don't leave out the definite and indefinite articles (the, a/an). Don’t write: “The new mechanism MUST allow for subsequent addition of new members.” Instead, write: The new mechanism MUST allow for *the* subsequent addition of new members.

3. Don't end sentences with prepositions (with, for, to, at), unless the result is really stilted. For example, don’t write: “I don’t like this group which you belong to.” Instead, write: “I don’t like this group to which you belong.” On the other hand, the following is correct, but very stilted: “Ending sentences with a preposition is a practice up with which I will not put.” Recasting such an awkward sentence to make it simpler is often the best solution. So instead write, “I will not put up with ending sentences with a preposition.” or “I will not put up with sentences that end with a preposition.”

4. Don't change subjects in mid paragraph.

5. Always spell check your specification before sending it out. Build a custom dictionary while you type, so that any special terms are included in the spell checking with the correct spelling.

6. Always proof-read your specification before sending it out for review.

7. Don't begin sentences with "This" as a noun referring to some unstated thing in the previous sentence. Only start sentences with “This” as an adjective with the noun immediately following. For example, "This section describes ..." is okay, but "This means ..." is not. Also avoid starting sentences with "There are".

8. Use adjectives to qualify nouns and thus be more specific. On the other hand, squeeze out excess verbiage that doesn't add any meaning; for example, question phrases like "are to be" and "is used to".

9. Instead of using “i.e.,” and “e.g.,” which mean that is and such as, respectively, use “that is” and “such as” (or “for example”), respectively. Many people don’t understand the subtle distinction between the two. The former means “the items are the exhaustive list and no other values are allowed” and the latter means “the items are examples but are not an exhaustive list so that other values are allowed as well”.

10. When two or more items are listed and then a parallel list is included, but each is intended to be pair-wise related, add the phrase: “, respectively.” For example, “Instead of using “i.e.,” and “e.g.,” use “that is” and “such as” (or “for example”), respectively.”