Developing a term checker

How do you write documentation to use ASD-STE100? Mike Unwalla decided to write his own software to control language usage.

Each day, my first task in the office is to develop my spatial awareness and manual dexterity skills by folding a page from my office calendar (www.amazingpaperairplanes.com/Calendar.html). Then, I start work.

My work has three parts:

- Writing documentation for customers
- Developing TechScribe as a business
- Developing a term checker, which I discuss in this article.

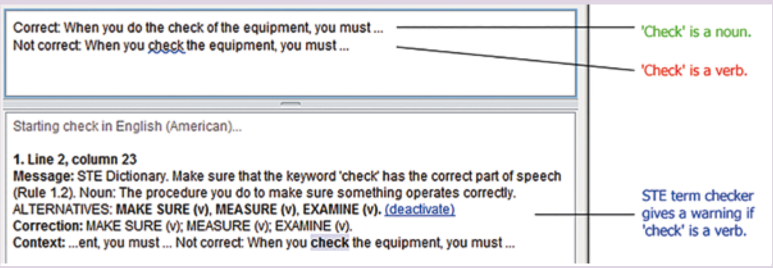
When I write documentation, terminology management is a large problem. For example, the preferred term for one customer is base *model deck*. Unfortunately, the existing documentation used 12 other synonyms. Initially, I was confused.

A controlled language helps to make text as clear as possible.

To make documentation clear, I use ASD-STE100 Simplified Technical English (www.asd-ste100.org). ASD-STE100 is a specification for the preparation of maintenance documentation in a controlled language. A controlled language helps to make text as clear as possible. For example, in ASD-STE100, the term *make sure* is approved, but synonyms such as *check*, *confirm*, *ensure*, *insure*, and *verify* are not approved. Some terms are approved only for one part of speech. For example, the term pump is approved as a noun, but not as a verb.

I wanted software to help me to make sure that text conforms to ASD-STE100, but I did not want to spend the many thousands of pounds that were necessary for commercial software. In 2008, an ISTC member wrote a Word macro that highlights different types of terms in different colours. The macro is excellent, but it gives no information about the problems that it identifies. For example, I know that a yellow term is not approved. However, I do not know whether an approved alternative exists. If an approved alternative does exist, I do not know what the alternative is.

In 2010, The Language Technologies Unit at Bangor University developed an online term checker (www.techscribe.co.uk/ta/ste2-term-checker-bangor-prototype.htm). Similar to Word macros, the online term checker is only lookup software.



Correct: When you do the check of the equipment, you must ...
Not correct: When you check the equipment, you must ...

Starting check in English (American)...

1. Line 2, column 23
Message: STE Dictionary. Make sure that the keyword 'check' has the correct part of speech (Rule 1.2). Noun: The procedure you do to make sure something operates correctly.
ALTERNATIVES: **MAKE SURE (v), MEASURE (v), EXAMINE (v).** (deactivate)
Correction: MAKE SURE (v); MEASURE (v); EXAMINE (v).
Context: ...ent, you must ... Not correct: When you **check** the equipment, you must ...

'Check' is a noun.

'Check' is a verb.

STE term checker gives a warning if 'check' is a verb.

Figure 1. If *check* is a verb, the STE term checker gives a warning

In 2011, I started to use LanguageTool (www.languagetool.org) to develop a term checker. Fortunately, I could use much of the example data from the Bangor University project. In January 2014, I released a commercial version of the term checker for ASD-STE100 issue 6.

LanguageTool gives a framework in which to create an effective term checker that can identify the part of speech that a term has. Rules for a language are in XML files. The rules specify patterns of text. If text is the same as a pattern, then LanguageTool gives a message. For example, in the text "the X was...," X is a noun. X cannot be a verb or some other part of speech. If the term *pump* is not in a pattern in which *pump* is a noun, then the term checker identifies the term as a possible error.

Sometimes, disambiguation is not possible. My favourite example is the sentence, "Operating systems that are slow can cause problems." In Simplified Technical English, the term *operating system* can be a technical name, but the verb *operating* is not approved. The sentence has two interpretations:

Correct: *Operating systems* that are slow can cause problems. (= If an operating system is slow, problems can occur.)

Not correct: *Operating* systems that are slow can cause problems. (= If you operate systems that are slow, problems can occur.)

If the term *operating system* is an approved term, you can add the term to the rules. The term checker will ignore the phrase *operating system*. However, the term checker cannot tell you whether your sentence has the correct meaning.

If a non-approved word or an unspecified word is in quotes or if it is part of a proper noun, ASD-STE100 lets you use the word. Examples from ASD-STE100: "Abort" button, Acceptance Test, Federal Aviation Administration, France.

To minimize the customization that is necessary, I wanted the term checker to ignore proper nouns. I struggled to create the rules because I was not sure what I wanted to do. Not all words that start with an uppercase letter are proper nouns. For example, the first word of a sentence starts with an uppercase letter. Also, I want to find incorrect text:

"When you Zero the meter, you must…"

"When you 'Zero' the meter, you must…"

Finally, I decided to have different rules. If text is in quotes, the term checker does not give a warning for non-approved words, but if the text is not in quotes, the term checker gives a warning.

To sell a product, a 'minimal marketable product' is necessary (www.romanpichler.com/blog/agile-product-innovation/the-minimal-marketable-product). The initial demonstration software had two large related problems and I did not have a product to sell:

- ASD-STE100 terms and the project terms were in the same XML file.
- The ASD-STE100 rules replaced the default rules that were in LanguageTool.

To update either LanguageTool or the ASD-STE100 rules was difficult. A solution is to keep the ASD-STE100 rules and the project terms in separate files outside the LanguageTool installation. Files in LanguageTool contain pointers to the files for the term checker.

Each time that I find an error in the disambiguation, I change the rules to remove the error and I put an example of the applicable text into the rules. (Each rule must have a minimum of one example of correct text and one example of incorrect text.) LanguageTool has a tool that makes sure that each rule gives the expected result.

Free version

A free demonstration version of the term checker for ASD-STE100 issue 3 is on www.simplified-english.co.uk
Let me know what you think about it.

References and resources

Aerospace and Defence Industries Association of Europe (ASD) Simplified Technical English
www.asd-ste100.org

Muegge, U (2013) 'TechScribe STE Term Checker' *Communicator*, Summer 2013: 28-29

Term checker for Simplified Technical English (ASD-STE100)
www.simplified-english.co.uk

Mike Unwalla FISTC
E: mike@techscribe.co.uk
W: www.techscribe.co.uk