Misunderstandings over the intended meaning of engineering specifications often result in growth of contract costs. Specification editors, therefore, should be especially interested in the phenomenon of ambiguity. This paper presents some useful doctrine borrowed from computational linguistics, and amenable to software implementation, that helps to systematize the detection process for two of the three classes of ambiguities. For the third class, it offers no such system, but gives brief advice on how the probability of detection may be enhanced.

About Specifications

Many engineers spend a large portion of their careers writing, interpreting, and meeting specifications. When in use, those specifications are read by many people with a variety of interests—interests perhaps never anticipated by the writers. Seen from so many different points of view, the specifications' words are sometimes erroneously taken to have meanings never intended by their writers. At other times, readers saddled with financial constraints may even be actively looking for ways to interpret the specifications differently from what they know the writer intended.

Specification Language

In this discussion, we are concerned neither with the subject matter covered in specifications, nor with the organization of that subject matter. Those topics are covered very well elsewhere. MIL-STD-490 (AFSC 1985) is a good example of such a source.

Our interest is in details of the language used in stating the requirements to be specified. While most sources on specification writing touch on the topic, only a few of the available works are outstanding in their treatment of the details of specification language. The more notable publications in this category are the Construction Specifications Institute’s (CSI) Manual of Practice (1983) and Henry Henkin’s book Drafting Engineering Contracts (1988). CSI tells us to state requirements in the active voice and imperative mood, thereby averting problems that arise from inordinately complex sentence structure. Active imperative specifications are much endorsed and widely used in private-sector construction work, but have not been widely accepted for use by public contracting agencies, where the traditional quasi-legal style is preferred (Pilfold 1987).

The Need for a Systematic Method

Everyone with significant experience at contracting has horror stories to tell of disputes that arose when specifications were misinterpreted. With bitter experience being so universal in the profession, it seems rather odd that no systematic methods have been proposed for examining specifications to predict such disputes before they happen. Engineers, who normally use analytical methods in everything they do, seem naively satisfied to do without a method of analysis that ensures the correctness of their specifications.

The traditional approach to checking specifications has been entirely intuitive. It consists of distributing the draft to as many reviewers as possible in hope that the document will be read many times over, and a large percentage of the errors found. The directions given to the reviewers are very general in nature, like "Look this over carefully to see if you can find anything wrong. Tell us what you find."

RT 9.5/168
Given so ill-defined a goal, it is not surprising that the review process adds very little value to the product. Even when we are more specific, and direct the reviewer to read each sentence and think of the ways in which it can be interpreted that are not to our advantage, we are asking the reviewer to perform an open-ended task. For such a process to work, we are depending on the reviewer to possess a vivid imagination and the special cognitive skill required to keep looking for meaning in the text long after one, two, three, or more possible meanings have been discovered (Dressel 1990).

While no purely analytical method for reviewing specifications exists, and it is likely that none ever will, we must press on at developing systematic, computerized methods, however crude those methods may be. The resources wasted as a result of errors missed by intuitive review techniques are too valuable.

A Starting Point

Another way of describing the open-endedness of the traditional review method is to borrow a term from computer science, and call it a “top-down” approach. The term, in this case, refers to the need for the reviewers to reason from abstract descriptions of defects towards specific instances in the specifications. Educators tell us that reasoning in such a manner is a task that humans generally do not perform well (Kline 1973), and we believe that staking the quality of specifications on such dubious performance is not wise. Engineers are not alone in this error; the futility of taking a top-down approach to rhetoric was very clearly described in 1936 by I. A. Richards as “poking the fire from the top,” long before the term “top-down” came into widespread use.

For an example of what “top-down” means, consider a type of specification error known as “agreement-to-agree.” If we tell the reviewers to find requirements for mutual agreements that must be reached at some future date, we can be almost certain that none will be flagged, even though the document examined may have numerous requirements for decisions to be made during planned design-review meetings. In this case, as in many others, the reviewers seem unable to connect the abstraction with the specific case. Better results could be had by telling the reviewers to flag all references to design-review meetings, and to flag a number of other phrases having similar, specific logical connections with agreements-to-agree. A similar process can be applied to other types of defects as well. In short, what we suggest is a simple automated “cookbook” that tells the reviewers, in bottom-up terms, what to look for and also helps them do the looking.

In this case, a simple search for the words “design review” will suffice. Our need, however, goes beyond what can be accomplished by simple lexical analysis. Specification defects may also be syntactic or contextual in nature.

Ambiguity

Linguists generally agree with the assertion that written English sentences may be ambiguous at three levels—usually called lexical, syntactic, and contextual. These categories are discussed in connection with specification writing by Henkin (1988).

Lexical difficulties result from the manner in which ideas and words do not map to each other on a one-for-one basis. The phenomenon is two-phased. On one hand, some ideas can be represented by several
interchangeable words, and on the other hand some words can be used to represent several different ideas. We all use the former in our informal writing to achieve "elegant variation." The latter is lexical ambiguity.

While on the topic of elegant variation, there is one brief point to make about its use in specifications. Elegant variation keeps our words from being monotonous, and writers of English use it habitually. Breaking ourselves of that habit when writing specifications is very important. Technical terms must always appear in exactly the same form every time a reference is made to the idea they represent (Henkin 1988). The requirement for clarity in specifications, as in certain other forms of technical writing, greatly outweighs the importance of elegance. Einstein has taught us an apt lesson on this point (1916).

"In the interest of clearness, it appeared to me inevitable that I should repeat myself frequently, without paying the slightest attention to the elegance of the presentation. I adhered to the precept of that brilliant theoretical physicist L. Boltzmann, according to whom matters of elegance ought to be left to the tailor and to the cobbler."

The pun in the last word was probably not intended by Boltzmann or Einstein, since, according to our German friends, there is no direct German equivalent to the English usage of "cobbler" in describing inept craftsmen. In this case, the pun also provides us with an example of lexical ambiguity. Here we see one of those rare instances where an ambiguous statement conveys two correct ideas. The normal effect of ambiguity is to convey an incorrect idea as well as the intended one. A quick look at any page of a dictionary will suffice to illustrate how lexical ambiguity pervades the English language. Only a handful of words may be found that have but one definition.

Syntactic ambiguity occurs when a single string of words can be mapped to multiple sentence structures. Dangling constructions are the most often described forms of syntactic ambiguity. Other types occur frequently in everyday language. Some are generally not recognized as erroneous, but can nonetheless result in misunderstanding.

Finally, in a contextual ambiguity, a single well-formed statement may apply correctly to more than one conceivable situation, even when the words take the same definitions and the syntactic structure is the same in each case. Contextual ambiguities exist because some necessary base of reference is incompletely defined by the context (Dressel 1990). Here is an example of contextual ambiguity in construction specifications:

*All surfaces shall be painted white to increase reflectance.*

In this case, a contractor believed that shiny metal surfaces were already more reflective than white paint, and therefore needed no paint. The Government wanted all surfaces painted, regardless. (GSBCA 1971).

**Finding the Lexical Ambiguities**

As stated above, what we need is a practical method to help reviewers with the ambiguity detection problem. Providing that method for lexical cases is quite easy. The first step is to make a glossary of ambiguous words and phrases that are often found in the documents being examined. In addition to ambiguous words and phrases, the
glossary may contain entries that are connected with other types of defects as well. For example, consider the word "asbestos," which denotes a substance we are no longer permitted to specify routinely.

Many publications on specification writing offer such a list of words and phrases, but none is comprehensive. Our list was compiled from numerous published and unpublished sources, and has been supplemented empirically. It presently has about 225 entries, many of which are peculiar to Government contracts for the purchase of training devices.

Memorizing so many words and finding every occurrence of them in a 75,000 word document is beyond human ability, but with help from a personal computer to do the searching, we can find every one. The seminal work on computerized inspection of text was reported by Kincaid (1981). A minor software industry has sprung up from those seeds, and offers a number of software packages that are usable for specification writing if supplemented with a suitably tailored rule set.

Our own software for flagging sensitive words and phrases was developed before the industry offered off-the-shelf products that were useful for our purposes. It has been in use for about three years, and has brought about a marked reduction in lexical ambiguities found in specification drafts.

Finding the Syntactic Ambiguities

A great deal of what we know today about ambiguity in the English language has been learned during the past forty years in the course of intensive study by computer science researchers. The aim of that study is to allow humans to communicate in their own natural language with computers, and the discipline is known as Natural Language Processing (NLP). Simultaneous exposure to both NLP and specification writing has led us to hypothesize that many of the same aspects of English syntax that complicate NLP may also play havoc with interpreters of specifications. We have tested that hypothesis by developing and exercising some rudimentary natural language processing software that does not attempt to resolve syntactic ambiguity, but merely reports the presence of a few types of syntactic structures that are notorious among practitioners of NLP (Read et al. 1990). Those types of syntax are illustrated in the examples given below.

- The phrase with multiple coordinating conjunctions:
  
  \ldots gluing and clamping or riveting.

Note that there is no fixed order of precedence of logical operators in English as there is in formal languages like Algol. This example is a very simple one. The really serious errors along these lines are to be found in long sentences that have several conjunctions.

- The string of prepositional phrases:
  
  \ldots predictions shall be prepared for use in the preliminary design-review meeting.

Here, the reader may prepare the predictions in the meeting, much to the consternation of the writer, who expected to use the predictions in the meeting. Notice that the meanings of the individual words are exactly the same regardless of which interpretation you prefer. The confusion arises because the reader has no clue as to which word is modified by each of the two prepositional phrases. Strings of prepositional phrases abound in specifications. Most are
disambiguated by semantic constraints. Those that remain ambiguous are very difficult to recognize without the help of hints from the computer program.

- The modifier that may or may not act across a conjunction:
  
  stainless steel nuts and bolts.

Specifications using this phrase require only that the nuts be of stainless steel. Bolts of any material would be acceptable, regardless of whether that is what the specification writer intended. Some linguists classify this as a case of ellipsis since the words “stainless steel” are intended to modify “bolts,” but their presence is assumed to be understood by the reader. Errors of this type are the most numerous syntactic ambiguities observed in specification drafts, and they deserve special attention. Formal languages solve this ambiguity problem by having operators that force the logical grouping of conjoined symbols. An example would be the begin . . . end grouping in Algol.

- Adjacent phrases that lack a clear boundary:
  
  The completed assembly shall be designed and constructed to withstand, without damage, permanent deformation, cracking or metal fatigue, stresses incident to movement, handling in transit, hoisting and tiedown aboard transporting vehicles, final installation, and use.
  
  The writer in this case probably intended a semicolon after “fatigue” but did not furnish it. That punctuation would have provided the necessary delimiter. Better form in so complex a situation would be to furnish the second noun phrase as a numbered, indented list. Punctuation often plays an essential role in disambiguating the syntax of sentences in specifications. Commas, hyphens, and semicolons must be used exactly according to the rules. Virgules (slash marks) are taboo.

- Ellipsis and gapping:
  
  The generator shall supply the processor with 10.5 amperes and the batteries 8.5 amperes.

  A “shall supply” has been left out, and there is no indication whether it belongs after “and” or after “batteries.”

- Ambiguous pronoun referents:
  
  Prior to accepting products from subcontractors, the prime contractor shall evaluate them for compliance with the standards.

  Which is to be evaluated, the products or the subcontractors? Flagging these in specifications is very easy. We simply flag every pronoun and direct the reader to make sure there is only one noun that may be attached to the pronoun.

In NLP, every occurrence of the structures described above constitutes a case of ambiguity that must be dealt with by NLP programs, sometimes with great difficulty. When one is found in specifications the sentence is not necessarily ambiguous, but the likelihood of ambiguity is greater than in sentences containing no such structure. The number of otherwise undetected ambiguities found with the help of the experimental software confirms our hypothesis. The ratio of items flagged to items requiring revision is high, but considering the potential cost of disputes over the interpretation of our specifications, we believe the software is a useful cognitive tool for finding ambiguities that are syntactic in origin.
Finding Contextual Ambiguities

Contextually ambiguous text has no easily detectable traits like the presence of certain words or of certain syntax. Its fault lies entirely in the realm of meaning. Therefore, until such time that computers are able to operate as well upon the set of meanings expressible in English as they are upon the set of meanings expressible in the language of logic and arithmetic, there will be no known way to provide automated hints as we have done for the lexical and syntactic cases. To detect contextual ambiguities, then, the best method we know is to assign the task to an editor who is truly trying to understand the specifications, not to someone who already understands them and is only reading them. The naive editor is naturally more likely to misunderstand the writer's intent, and detecting such possibilities, after all, is our objective.

AIR FORCE SYSTEMS COMMAND (AFSC)
MIL-STD-490A
Washington D. C. 1985

CONSTRUCTION SPECIFICATIONS INSTITUTE
Manual of Practice
Washington, D. C. 1983

DRESSEL, Susan
Reference Bases in Natural Language and Technical Language
Communication Across the Sea: North American & European Practices. Conference Record
IEEE Professional Communication Society, September 1990

EINSTEIN, Albert

GENERAL SERVICES BOARD OF CONTRACT APPEALS (GSBCA)
Robert E. McKee General Contractor, No. 3175, 71-1 BCA 8625
Contract Appeals Decisions
Chicago: Commerce Clearing House 1971

HENKEN, Henry
Drafting Engineering Contracts

KINCAID, J. Peter et al
Implementing the Computer Readability Editing System (CRES), TAEG Report No. 98

KLINE, Morris
Why Johnny Can't Add
New York: St. Martin's Press 1973

PILFOLD, David
Legal Style in Specifications

READ, Walter et al.
Natural Language Sourcebook, Project Report #13
Los Angeles: UCLA Center for the Study of Evaluation 1990

RICHARDS, Ivor A.
The Philosophy of Rhetoric
Oxford: Oxford University Press 1936