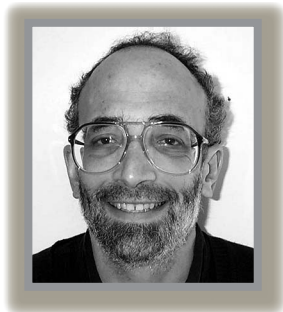# quality time

The Syntactically Dangerous *All* and Plural in Specifications

Daniel M. Berry and Erik Kamsties

We're academics. When each of us wishes to augment his income with consulting, his favorite positions are requirements specification (RS) inspector or inspection facilitator. Each of us has developed his own private check list of potential-problem indicators, and, when he inspects an RS, he reports any instances of these indicators to the authors. Often enough, these potential problems prove to be actual ones.

Among the items on both of our lists are specific problems involving the correct use of the natural language in which the RS was written, usually English or German for our clients. These issues include incorrect grammar, incorrect word placement, and all kinds of ambiguities.[1,2] Perhaps surprisingly, our lists of grammatical and word-placement problems are similar despite the difference in the natural languages involved. These syntactic problems are symptoms of ambiguities in meaning—a grammatical problem occurs when part of a sentence disagrees with another, and each choice in the disagreement corresponds to a different meaning. (From our common background in compiling, contrary to the usage in linguistics,[3] we use *syntactic* to classify sentence-form issues and *semantic* to classify sentence-meaning issues.)

We'll discuss only one such class of problems, apparently not treated in the software engineering literature: the syntactically dangerous *all* and plural in specifications. The literature has covered other issues, including the dangerously misplaced *only* (thoroughly discussed by Peter Neumann[4]) and the semantically dangerous *all* (thoroughly discussed by Chris Rupp and Rolf Goetz[5] and by us[6]).

The problem

One of us, Daniel Berry, discovered the issue we address while facilitating an inspection of an RS. We've modified the example to protect the client's identity and trade secrets, but the example is structurally identical to the one that occurred during the inspection. Consider the sentence

All the lights in any room have a single on-off switch.

How many switches does any room have, one or one per light? Berry noticed that some parts of the RS appeared to assume that all the lights in any given room share a single on-off switch. Other parts of the same RS appeared to assume that each light in any room has its own on-off switch. From Berry's limited domain knowledge, each choice seemed equally

plausible; he knew of rooms one way and of rooms the other way. Moreover, when Berry asked which choice was correct during the inspection meeting, some members of the team had assumed one way, some had assumed the other, and the rest hadn't even thought about the issue.

Berry realized that this example was but one of a general problem. Later, when Berry was supervising Kamsties's PhD research on ambiguities in RSs, we thoroughly investigated the example. We named it the "syntactically dangerous *all* and plural in RSs" problem.

Syntactic danger of *all*

The problem with the example given earlier is that it's unclear whether

1. each light in a room has its own single on-off switch,

or

2. all lights in a room share a common single on-off switch.

Either of the following sentences clearly conveys meaning number 1:

> *Each light in any room has a single on-off switch.*

> *Each light in any room has its own on-off switch.*

The following sentence clearly conveys meaning number 2:

> *All lights in any room share a single on-off switch.*

Scopes and domain knowledge

The sentence we examined,

> *All the lights in any room have a single on-off switch.*

is a classic example of scope ambiguity. It's unclear which quantifier equivalent—*all*, for "∀," or *a*, for "∃!" (there exists a unique)—takes precedence over the other. Mathematics clearly shows the problem. The two meanings of this sentence are

> ∀*y* ∈ *the lights in a room*, ∃!*x such that x is the on-off switch of y*

> ∃!*x such that* ∀*y* ∈ *the lights in a room, x is the on-off switch of y*

Many times, this ambiguity is hidden by domain knowledge. For example, consider the structurally similar ambiguous sentence:

> *All persons have a unique national insurance number.*

Domain knowledge tells the reader that the intended meaning of the sentence is that each person has his or her own unique national insurance number, and not the ridiculous idea that all people share a common unique national insurance number. So ridiculous is the second option that most readers wouldn't even think of the other possible meaning of the sentence and wouldn't think the sentence ambiguous.

By the way, this example illustrates the semantically dangerous *all*. This sentence is dangerous because it's probably not true; a person might have zero, two, or more national insurance numbers, and a program that assumes that each person has exactly one is sure to fail.

Problem with plural

The use of plural to describe a property of elements of a set or of sets makes it difficult to determine whether the property is that of each element or of the whole set. Consider the two structurally identical sentences:

> Use "each" when the intention is to talk about properties of each member of the set, and use "all" when the intention is to talk about properties of the whole set.

> *Students enroll in six courses per term.*

> *Students enroll in hundreds of courses per term.*

Domain knowledge tells us that the first sentence is talking about each student while the second is talking about the whole set of students. Without this domain knowledge, nothing in either sentence indicates whether enrollment in the stated number of courses per term is a property of each student or of the set of all students.

Because the first sentence is talking about each student, it should be written in singular form, as one of

> *Each student enrolls in six courses per term.*

> *A student enrolls in six courses per term.*

The first is more for stating the fact that *every* student enrolls in six courses per term, and the second is more for specifying a rule that a student should enroll in six courses per term. Using a singular formulation for talking about properties of each or any student reserves the plural formulation for talking about properties of the collection of students:

> *Students enroll in hundreds of courses per term.*

Other languages

These syntactic problems with plural universal quantifier equivalents and with plural sentences aren't restricted to English. For example, each of the previous examples using *all* or *each* can be duplicated with the same meanings in French with *tous* or *chaque*, in German with *alles* or *jeder*, in Hebrew with *col* or *col ekhad*, in Italian with *tutti* or *ogni*, and in Portuguese and Spanish with *todo* or *cada*, respectively.

With adjectives, for example, in the sentences

> *All humans are mortal.*

> *Each human is mortal.*

the problem might seem not severe,

since the same adjective works for both singular and plural, and mortality is a property that holds equally well for the individual or for the whole set. However, in some languages other than English, an adjective for a singular noun is different from the same adjective for a plural noun. For example, in French:

Tous les hommes sont mortels.

Chaque homme est mortel.

Interestingly, mathematics has adopted a convention that makes intent clear. In mathematics, the universal quantifier ∀, read as *for all*, is singular as in,

$$\forall y \in \text{Int}, \; x < x + 1$$

For all integers x, x is less than x + 1.

The same syntactic problem exists with other, nonuniversal quantifier equivalents—for example, *some* and *many*, which are all plural. You're left with the inability to specify what's true of each of the some or each of the many without resorting to constructions such as *many a person*, *each of many persons*, *each of some persons*, and so on.

The simplest way to write a sentence describing a property of each member of a set is to use *each*, which is grammatically singular. When you use *each* in or as the subject of a sentence, the verb of the sentence must be singular. Consequently, also the complement of the sentence must be singular. When the complement is singular, the grammatical implication is that the complement applies to each member of the set. Thus, the simplest way to be precise about intention is to use *each* when the intention is to talk about properties of each member of the set and to use *all* when the intention is to talk about properties of the whole set.

A specification inspector can certainly search for plural constructions in a specification to examine each for its danger. Best of all is for a specification writer not to write plural statements when describing properties of each element of a set.

Finally, you should notice our studious avoidance of the use of plural in this article to describe general situations. Indeed, with respect to the dangerously misplaced *only* and the semantically dangerous *all* problems, you should notice that in this article, each occurrence of *only* (and similar words such as *also*) and each occurrence of *all* (and similar words such as *each*) has been used correctly according to the recommendations of the literature we've cited. ⑤

References

1. D.M. Berry, and E. Kamsties, "Ambiguity in Requirements Specification," *Perspectives on Requirements Engineering*, J.C.S.P. Leite and J. Doorn, eds., Kluwer, 2004, pp. 7–44.
2. D.M. Berry, E. Kamsties, and M.M. Krieger, *From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity*, tech. report, Computer Science Dept., Univ. of Waterloo, 2003; http://se.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf.
3. J. Allen, *Natural Language Understanding,* 2nd ed., Addison-Wesley, 1995.
4. P.G. Neumann, "Only His Only Grammarian Can Only Say Only What Only He Only Means," *ACM SIGSOFT Software Eng. Notes*, vol. 9, no. 1, 1986, p. 6.
5. C. Rupp and R. Goetz, "Linguistic Methods of Requirements-Engineering (NLP)," *Proc. European Software Process Improvement Conf.* (EuroSPI), Copenhagen Business School, 2000; www.iscn.at/select_newspaper/requirements/sophist.html.
6. D.M. Berry and E. Kamsties, "The Dangerous 'All' in Specifications," *Proc. 10th Int'l Workshop Software Specification and Design* (IWSSD-10), IEEE CS Press, 2000, pp. 191–194.

Daniel M. Berry is a professor of computer science and software engineering at the University of Waterloo in Canada. Contact him at dberry@uwaterloo.ca.

Erik Kamsties is a senior research assistant in the Institute for Computer Science and Business Information Systems at the University of Duisburg-Essen in Germany. Contact him at kamsties@sse.uni-essen.de.