The Dangerous "All" in Specifications

Daniel M. Berry, CSD, University of Waterloo, Canada

Erik Kamsties, Fraunhofer IESE, Germany

as the source of very common assumptions and exceptions arising from these assumptions.

Specifications, CBSs, and REs

We are talking about specifications of computer-based systems (CBSs) and, especially, of the software components of them.

These specifications are written by requirement engineers (REs).

This will prove related to $D,S \vdash R$!!!

Dangerous Sentences

Christine Rupp and Rolf Götz, in "Sprachliche Methoden des Requirements Engineering" caution specifiers of the dangers of using universal quantifier equivalents, e.g.,

"never",
"always",
"none",
"each",
"all", etc.,

in natural-language specifications.

Not Just in N-L Specifications

Actually, the danger is also in formal specifications.

The Danger

A statement involving such quantifier equivalents is sometimes dangerous because it may simply *not be true*.

For a CBS to assume that it is true is courting disaster when an unanticipated input comes along and the CBS is not prepared to respond to it gracefully.

Example Specification:

“Each person has a unique national insurance number.”*

*** Most likely, one would say, “All persons have a unique national insurance number”, but that is not correct for reasons discussed later.**

Mostly True

This statement is “mostly true”, “occasionally false”, and thus logically false.

There are persons who for one reason or another have gotten more than one number.

For a national insurance CBS to assume that each person has precisely one number is downright dangerous.

Must Deal with Anomalies

The CBS must deal with all sorts of anomalies, including, that a given person

- **has more than one number,**
- **has never been assigned a number,**
- **reports an invalid number, and**
- **reports someone else's number,**

whether maliciously or accidently.

There may be other anomalies not listed here.

Other Dangerous Words

Similar examples can be written involving other universal quantifier words such as

“never”,
“always”,
“none”, and
“all”.

Not Always Dangerous

However, there are times in which such strong universally quantified statements are appropriate.

A robust procedure should be able to handle all inputs, even if the mathematical function it implements is undefined for some inputs.

For input not in the function's domain, the procedure should at least report that the input is illegal.

D

When Dangerous and When Not?

When are universally quantified statements dangerous and when are they not?

Notions offered by Michael Jackson and Pamela Zave provide the distinction.

These are the same MJ and PZ from the World model and $D,S \vdash R$!!!

Descriptions and Requirements

Jackson and Zave talk about

- ***descriptions*** **of (*domains* or *real worlds*), and**
- ***requirements*** **or *problems*.**

Domains

“The domain is the subject matter of the system’s computations, and provides the context in which those computations have useful meaning or effect.”

A domain is “a topic for description in its own right, independently of any description that we may eventually make of the system to be constructed.”

Two Kinds of Sentences

Jackson and Zave divide sentences in a specification into two classes, those that describe the domain and those that describe requirements.

These are in two different grammatical moods, *indicative* and *optative*.

Indicative and Optative Moods

1. **A sentence about the domain is in the *indicative* mood, asserting truths about the domain, describing the world as it is, independent of any computation placed in it.** D

2. **A sentence about the requirements is in the *optative* mood, describing what the computation being specified is required to bring about, describing the world as it will be after the specified computation is placed in it.** R

Indicative Example

“Each person has a unique national insurance number.” D

is an attempt to be an indicative statement about the real world.

It is incorrect!

It is clearly independent of any computation that we might wish to impose on the real world.

Indicative Example Corrected

“Except for exceptions described elsewhere, each person has a unique national insurance number.”

Optative Example

“The national insurance system shall deal with each input that is claimed to be a national insurance number.”

R

This sentence is an optative statement about a CBS to be built in the real world.

Distinction Defines Danger

With this distinction, it is clear when universally quantified statements are dangerous and when they are not.

Indicative Danger

A universally quantified indicative statement is dangerous because ...

it probably is not true.

Assuming that it is true leaves the CBS unable to deal with all possible inputs.

More Indicative Danger

Universally quantified indicative statements lull CBS designers into not investigating all possible contingencies.

An RE who believes the customer's claim that "Each person has a unique national insurance number." is less likely to investigate all the possibilities

He/she is less likely to discover the exceptions mentioned above, with which the CBS must deal.

Some Exceptions to Rule

There are universally quantified indicative statements that are true.

“Each human is mortal.”

However, such statements are rare.

Caveat

In general, each universally quantified indicative statement has to be examined closely to search for exceptions or to ascertain that it is indeed true.

D

Optative Striving

On the other hand,...

R

a universally quantified optative statement is reasonable and often desired.

Optative Striving Example

It is reasonable to require that the national insurance CBS deal with each input claiming to be a national insurance number.

The CBS should be able to handle the four exceptions mentioned above...

as well as the normal case, in which the number belongs to one and only one person.

Handle Even Surprises

The CBS should be able to handle also any situation that has not been thought of and described in the specifications.

Conclusion

A specification consists of two kinds of sentences,

- **indicative and** D
- **optative.** R

Red Flag

A universally quantified indicative statement is probably not true.

D

It should thus raise a red flag.

It should be a signal to the REs to ask when it might not be true, to allow discovery of all the exceptions that must be handled.

Challenging Goal

A universally quantified optative statement is a challenging goal for all (note the universal quantifier in this essentially optative statement) CBSs.

R

It indicates the goal that each CBS handle both its normal cases and all possible exceptions and contingencies.

Yet Another Signal

A universally quantified optative statement should be yet another signal to the REs to search for other contingencies that the CBS should handle. R

More Danger for “All”

What *is* the problem with:

“All persons have a unique national insurance number.”?

Grammar Problem

“All” is plural.

As written, “All persons have a unique national insurance number.” means that all persons share a unique national insurance number.

To avoid that meaning and still use “all”, one would have to write “All persons have unique national insurance numbers.”.

Meaning Problem

But, it is not clear that “unique” can be used with a plural noun.

So then write “All persons have national insurance numbers.”

But then, it is not clear how many numbers there are per person.

Avoiding Problem

This problem is avoided by using the singular "each".

"Each person has a unique national insurance number."

It is clear that the association of persons and numbers is 1–1.

Acknowledgments

The authors thank Jo Atlee, Don Cowan, and Michael Jackson for their comments on earlier drafts of the related paper.

References

1. **C. Rupp and R. Götz, “Sprachliche Methoden des Requirements Engineering”, Technical Report, SOPHIST GmbH, Nürnberg, Germany, 2000**

References, Cont'd

2. **M. Jackson and P. Zave, "Domain Descriptions", *Proceedings of the International Symposium on Requirements Engineering*, IEEE Computer Society, 1993, pp. 56–64.**
3. **P. Zave and M. Jackson, "Four Dark Corners of Requirements Engineering", *ACM Transactions on Software Engineering and Methodology*, 6: 1, pp. 1–30, 1997.**